# EDGE-BASED IMAGE INTERPOLATION USING SYMMETRIC BIORTHOGONAL WAVELET TRANSFORM

by

WEIZHONG SU

### A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

#### THE FACULTY OF GRADUATE STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

September 2005

© Weizhong Su, 2005

### ABSTRACT

Image interpolation is an important part of digital image processing. Many approaches are proposed for enlarging and reducing images. Recently, most papers in image interpolation are focused on edge-based interpolation since sharp edges and smooth contours can give better impression to the human vision than others. Li & Orchard and Kimmel proposed edge-based interpolation approaches that can produce better image quality compared with the traditional methods such as bilinear and bicubic interpolations. In this thesis, a new edge-based image interpolation approach that uses symmetric biorthogonal wavelet transforms is proposed.

According to wavelet multiresolution analysis theory, an image can be decomposed into a series of approximation sub-images and detail sub-images with horizontal, vertical, and diagonal edge information. Based on this theory, many wavelet-based interpolation approaches have been proposed. However, most of them are computationally expensive or not efficient. In this thesis, we set up a list of ideal step edge models, and explore the relationships between the wavelet approximation sub-image and the three wavelet detail sub-images of these models. Based on these relationships, a fast and efficient algorithm that predicts the edge information of the interpolated image is proposed. The results of our experiments prove that the wavelet-based image interpolation approaches. In conclusion, the 9/7-M inverse wavelet transform with our new approach is the best solution for image interpolation.

Keywords: edge-based image interpolation, wavelet transform.

# TABLE OF CONTENTS

.

.

ABSTRACT	ii
TABLE OF CONTENTS.	iii
LIST OF TABLES	V
LIST OF FIGURES	vi

CHAPT	ER 1 INTRODUCTION	
1.1	What Is Image Interpolation?	1
1.2	The Origins of Image Interpolation	2
1.3	Applications of Image Interpolation	
1.4	Existing Problems in Image Interpolation	4
1.5	Overview and Contribution of this Thesis	7
СНАРТ	ER 2 PRELIMINARIES	8
2.1	Introduction	8
2.2	Review of Image Quality Measurement	8
2.2.	.1 Introduction to Image Quality	8
2.2.	.2 Full-Reference Image Quality Measurement	11
2.2.	.3 Blind Image Quality Measurement	14
2.3	Review of Wavelet Transforms	16
2.3.	.1 Background	16
2.3.	.2 Multiresolution Analysis	18
2.3.	.3 Lifting Scheme	
СНАРТ	ER 3 STATE OF THE ART IN IMAGE INTERPOLATION	31
3.1	Introduction	
3.2	Conventional Approaches	
3.2.	.1 Bilinear Interpolation	
3.2.	.2 Bicubic Interpolation	
3.3	Edge-Directed Interpolation	35
3.3.	.1 Kimmel's Approach	

		iv
3.3	2 Li & Orchard's approach	37
СНАРТ	ER 4 PROBLEMS OF WAVELET-BASED IMAGE INTERPOLATION	41
4.1	Introduction	41
4.2	Problems of Wavelet Transforms	43
4.2	.1 Approximation Issues	43
4.2	.2 Magnification Issues	46
4.3	Problems in Approaches Using Mallat's Theory	49
4.4	Summary	52
СНАРТ	<b>TER 5 NEW WAVELET-BASED IMAGE INTERPOLATION APPROACH</b>	53
5.1	Edge Models of Images	53
5.2	New Image Interpolation Approach	59
5.3	Implementation	63
5.4	The Performance of the Proposed Approach	64
5.4	.1 The Gain of the Proposed Approach	64
5.4	.2 Comparison of Interpolation Approaches for Image Restoration	69
5.4	.3 Comparison of Interpolation Approaches for Image Magnification	73
5.6	Computation Time	76
5.5	Summary	77
СНАРТ	ER 6 CONCLUSIONS AND FUTURE RESEARCH	78
6.1	Conclusions	78
6.2	Future Research	79
BIBLIC	OGRAPHY	80
APPEN	DIX A: TEST IMAGES	85
APPEN	DIX B: RESTORED IMAGES (CAMERAMAN)	91
APPEN	DIX C: MAGNIFIED IMAGES (CAMERAMAN)	102

# LIST OF TABLES

Table 2.1	Some wavelet transforms using the lifting scheme24
Table 2.2	Parameters of some wavelet transforms25
Table 2.3	Computational complexity of some wavelet transforms25
Table 4.1	Performance comparison of wavelet transforms using approximation coefficients
	(PSNR)44
<b>Table 4.2</b>	Performance comparison of wavelet transforms using approximation coefficients
	(MSSIM)45
Table 4.3	Average gains of wavelet transforms using approximation coefficients (reference:
	bicubic)45
Table 4.4	Image enlargement performance of wavelet transforms (IQ)47
Table 4.5	Image enlargement performance of wavelet transforms (Contrast)48
Table 4.8	The wavelet extrema for different signals50
Table 4.9	The non-symmetric wavelet extrema51
Table 5.1	Gradients of images having ideal step edges55
Table 5.2	Gradients of wavelet-reduced images having ideal step edges55
Table 5.3	Extrema patterns and subtypes of wavelet-reduced ideal step edges56
Table 5.4	The profiles of a step edge with 30-degree direction
Table 5.5	The performances of new approach for wavelet-reduced images (PSNR: dB)65
Table 5.6	The performances of new approach for wavelet-reduced images (MSSIM)65
Table 5.7	The performances of new approach for original images (IQ)68
Table 5.8	The performances of new approach for original images (Contrast)
Table 5.9	The performances of several interpolation approaches for original images (IQ,
	×2)74
Table 5.10	The performances of several approaches for original images (Contrast, ×2)74
Table 5.11	Comparison of the computation time for different interpolation methods77

# LIST OF FIGURES

Figure 1.1	Some problems caused by the image interpolation	6
Figure 2.1	Aliasing	9
Figure 2.2	Blurring	10
Figure 2.3	Ringing	10
Figure 2.4	Blocking	10
Figure 2.5	Diagram of the structural similarity (SSIM) measurement system	13
Figure 2.6	Illustration of the quality map from the MSSIM measurement	14
Figure 2.7	Power spectrum	15
Figure 2.8	Several different families of wavelets	17
Figure 2.9	Scaling and wavelet function spaces	18
Figure 2.10	A two-stage or two-scale FWT analysis bank and its frequency-splitting	
	characteristics	20
Figure 2.11	The two-dimensional fast wavelet transform	21
Figure 2.12	Lifting scheme	22
Figure 2.13	5/3 filters	26
Figure 2.14	9/7-M filters	26
Figure 2.15	5/11-C filters	27
Figure 2.16	2/10 filters	27
Figure 2.17	9/7-F filters	28
Figure 2.18	5/3 scaling and wavelet functions	28
Figure 2.19	9/7-M scaling and wavelet functions	29
Figure 2.20	5/11-C scaling and wavelet functions	29
Figure 2.21	2/10 scaling and wavelet functions	30
Figure 2.22	9/7-F scaling and wavelet functions	30
Figure 3.1	Linear interpolation	32
Figure 3.2	Illustration of bilinear interpolation	32
Figure 3.3	Cubic interpolation	34
Figure 3.4	Illustration of bicubic interpolation	34
Figure 3.5	Pixel interpolation in Kimmel's approach	36
Figure 3.6	Geometric duality	38
Figure 3.7	An example of Li & Orchard's algorithm	39

Figure 4.1	Wavelet transform illustration
Figure 4.2	A wavelet-based interpolation problem model42
Figure 4.3	Performance comparison of wavelet transforms using approximation coefficients
0	(strong edges)46
Figure 4.4	Image enlargement performance of wavelet transforms (strong edges)
Figure 4.5	The propagation of the wavelet extrema
Figure 4.6	The decay of wavelet extrema
Figure 5.1	Ideal step edges with different directions
Figure 5.2	Block diagram of new image interpolation approach
Figure 5.3	The code snippet of the prediction for a 30-degree, subtype I edge60
Figure 5.4	The influence of the neighbors of extrema on an image expanded by a factor 862
Figure 5.5	Addition operations for crossed edges63
Figure 5.6	Wise edge selector
Figure 5.7	The performances of new approach for the strong edges from a wavelet-reduced
	image66
Figure 5.8	The performances of new approach for the strong edges from an original
	image67
Figure 5.9	Several low-pass filters
Figure 5.10	Comparison of interpolation approaches using different low-pass filters (PSNR,
	average)70
Figure 5.11	Comparison of interpolation approaches using different low-pass filters (MSSIM,
	average)70
Figure 5.12	Comparison of interpolation approaches using different low-pass filters (PSNR,
	'Cameraman')71
Figure 5.13	Comparison of interpolation approaches using different low-pass filters (MSSIM,
	'Cameraman')71
Figure 5.14	Restored 'Cameraman' images using different interpolation methods72
Figure 5.15	Performance comparisons of restored strong big edges73
Figure 5.16	Several areas from 2x Enlarged 'Cameraman' image75
Figure 5.17	16x enlarged 45-degree step edge images using different interpolation
	approaches76

vii

# INTRODUCTION

# 1.1 What Is Image Interpolation?

We have all seen movies in which FBI agents enlarge and enhance an image so that faces can be identified and so on... well, that is rather exaggerated, but image interpolation is something like that. It's not going to produce features that weren't there before, but what it does do is to maintain features that originally existed rather than make the image blurry when it's scaled up.

So, what is interpolation? One concise definition in *Merriam-Webster's* dictionary is "estimating values between two known values." A precise answer from Thévenaz et al. [16] is "model-based recovery of continuous data from discrete data within a known range of abscissa." From the mathematical point of view, interpolation seeks to define mathematical functions that pass through given data points. And image interpolation is just a two-dimensional case.

Traditional interpolation can be expressed with the formula below:

$$f(x) = \sum_{k} f_k \varphi(x - k) \tag{1.1}$$

where  $f_k$  are the values of known points, and  $\varphi(x-k)$  is a weighting function. The formula (1.1) means that an interpolated value f(x) is a linear combination of the weighted values of known points. The constraint of the formula (1.1) is that  $f(k) = f_k$ . Therefore, the traditional interpolating function (1.1) passes through all known points.

Thévenaz et al. [16] provide a generalized interpolation formula:

$$f(x) = \sum_{k} c_k \varphi(x - k) \tag{1.2}$$

where, coefficients  $c_k$  depend on  $f_k$ , but generally are not equal  $f_k$ . This means the interpolating function (1.2) no longer passes through all known points. This generalized formula

gives us more freedom to choose a weighting function with better properties than those available in the restricted classical case where  $c_k = f_k$ .

# **1.2 The Origins of Image Interpolation**

Interpolation dates back to the Seleucid period (the last three centuries BC) when ephemerides were used to predict the position of the sun, moon, and planets to plan the best dates for events such as planting crops [7]. Because celestial bodies were not always in position to be observed, linear interpolation and more complex interpolation methods were used to fill gaps in the ephemerides.

In the early 1970s, with the advance of computer technology, digital image processing started to develop. Simple interpolation methods generally cannot produce a high-quality image. For examples, the pixel replication yields blockiness, and linear interpolation tends to blur the image. To get a better quality image, high-order interpolation schemes were introduced. For example, a very popular interpolation technique known as cubic convolution was used to improve satellite images. Key's cubic convolution scheme [14] was based on this technique, and has become a standard in the field.

Spline interpolation for digital image interpolation was first investigated in 1978. Spline interpolation is based on the generalized interpolation formula (1.2). At first, people thought that spline interpolation would need more computations because of the cost of determining the  $c_k$  in the formula (1.2). In the early 1990s, Unser showed that using a digital-filtering approach, which can be implemented recursively, could solve a B-spline interpolation problem much more efficiently. In 2003, Blu et al. [15] showed that shifted linear interpolation had almost the same performance as cubic convolution. Also, recent studies show that spline interpolation provides the best cost-performance tradeoff over other conventional methods [17].

In the last decade, image interpolation methods based on discrete cosine transforms (DCT) and wavelet transforms began to emerge. These transform-based image interpolation techniques cause artifacts stemming from the specific properties of the transform used. Therefore, they didn't receive much attention. More detailed information about wavelet-based image interpolation will be presented in chapter 5.

Recently, published papers focusing on edge-based image interpolation are appearing. This is because clear edge contours give a better visual effect. Allebach and Wong present the first

Chapter 1 Introduction

generally recognized paper [21]. They use a sub-pixel edge estimation technique to generate a high resolution edge map from a low resolution image, and then use the high resolution edge map to guide the interpolation of the low resolution image to a final high resolution version. The two most famous approaches to edge-based image interpolation are Kimmel's and Li & Orchard's. In Kimmel's approach [22], the likelihood that each pixel belongs to an edge is calculated based on the directional derivatives. Then the interpolation is performed such that more weight is given to the pixels lined up along the edge than the pixels across the edge. Li & Orchard's approach [23, 24] is based on the assumption that the high-resolution covariance is the same as the low-resolution counterpart. It's called geometric duality. Muresan and Parks' approach [28, 29] is an extension of Li & Orchard's approach, and is based on the optimal recovery theory. Chapter 3 of this thesis will present Kimmel's and Li & Orchard's methods in detail.

## **1.3 Applications of Image Interpolation**

Image interpolation is used extensively in digital image processing. Currently, the most important application areas are medical imaging, computer vision, and digital photography.

Image interpolation methods occupy an important position in medical image processing, as they are required for image generation as well as image post-processing. In computed tomography (CT) or magnetic resonance imaging (MRI), image reconstruction uses interpolation techniques to approximate the discrete functions in the back-projection for inverse Radon transform. In modern X-ray imaging systems, such as digital subtraction angiography, interpolation enables the computer-assisted alignment of the current radiograph and the mask image. In volumetric imaging, scene-based and object-based methods are commonly used for slice interpolation of three-dimensional (3-D) medical data sets. The purpose of the interpolation is to change the level of discretization (sampling) of the input scene. Interpolation is also necessary in the following situations: (1) to change the nonisotropic discretization of the input scene to an isotropic or to a desired level of discretization; (2) to represent longitudinal scene acquisitions of a patient in a registered common coordinate system; (3) to represent multimodality scene acquisitions in a registered coordinate system; and (4) in re-slicing the given scene at a different slice orientation [5]. In volume rendering, it is common to interpolate a texture to fit the facets that compose the rendered object. In addition, volume rendering may require the computation of gradients, which is best done by taking an interpolation model into account. Moreover, zooming or rotating medical images after their acquisition is often used in

#### Chapter 1 Introduction

diagnosis and treatment, and interpolation methods are incorporated into systems for computeraided diagnosis, computer-assisted surgery, and picture archiving and communication systems.

In computer vision, the computer relies on interpolation to display text and graphics correctly. For example, before a letter "A" shows on the screen, the computer takes dot matrix or TureType curves (quadratic B-splines) from the font library, reduces/enlarges the font according to the required size, and then shows it on the monitor screen. Image viewing/editing programs use interpolation when rotating, resampling, or resizing images. In the image editing programs, a user has some controls over the type of interpolation being used. There are a variety of interpolation techniques that may be used for such functions as enlarging or reducing the size of an image. For example, Adobe's Photoshop program allows the user to choose from one of three techniques: nearest neighbor, bilinear, and bicubic.

In recent years, digital cameras have become popular. Due to the limited spatial resolution of the physical structure of the color CCD or CMOS sensors, one kind of interpolation algorithm, called demosaicing, is used to reconstruct a super-resolution image beyond the sensor's resolution. Meanwhile, users can choose a different resolution for the saved image that also needs interpolation to resample the acquired image.

In addition to these applications, good insights into image interpolation techniques also help to develop better tools in such areas of image processing as image compression and image denoising.

### **1.4 Existing Problems in Image Interpolation**

The issue of quality is particularly relevant to the medical community; for ethical reasons, it is a prime concern when manipulating data. Any manipulation should result in the least amount of distortion or artifacts, so as not to influence the clinician's judgment. For practical reasons, efficiency is another prime concern. Any processing should result in the least computational effort, particularly when dealing with the large amounts of data involved in volumetric medical imaging. Therefore, to evaluate an image interpolation scheme, it is important to assess the computation time of the algorithm and the quality of the interpolated image. Different applications will have different criteria. Real-time applications may emphasize the computation time; and non-real-time applications may emphasize image quality more.

Let us examine traditional image interpolation techniques and discuss their advantages and shortcomings. These techniques are known as the nearest neighbor, the bilinear, and the Bicubic methods.

Strictly speaking, the nearest neighbor, which is also called replication, is not an interpolation algorithm. The new pixel's value is made the same as that of the closest existing pixel. The nearest neighbor is quite a bit faster than the bilinear or bicubic methods, but it is also the least precise. When enlarging images, edges are more noticeably jagged. Downsizing produces a coarse, grainy effect. It works best when enlarging or reducing by an even number.

Bilinear resampling is an interpolation method that uses the values from the four surrounding pixels that is, above, below, right, and left of the spot where the new pixel is to be created. The new pixel value is determined by calculating the weighted average of the four closest pixels (a 2x2 array) based on distance. This method tends to make an image appear softer; that is, the contrast is reduced because of the averaging of neighboring values. However, the stair-step effect apparent in the nearest neighbor approach is reduced, and the image looks smoother. This interpolation method appears to work better for image reduction rather than image enlargement.

Bicubic interpolation determines the values of new pixels by calculating the weighted average of the closest 16 pixels (a 4x4 array) based on distance. Usually, a cubic B-spline algorithm is used. This method also produces a much smoother image than the nearest neighbor technique. As with bilinear resampling, bicubic interpolation tends to make an image softer. So it is a good idea to apply some sharpening to the image to reduce the softness. Some bicubic algorithms include an extra parameter that you can use to sharpen image quality during the interpolation process. Although this method requires more computational time, it is the default image-enlargement technique in just about every image manipulation program.

Therefore, researchers have been working hard to find better methods of image interpolation. Some methods do improve the quality of images, but they can be used only on specific images or they introduce some artifacts. No perfect solution yet exists. Especially when large magnifications are required, the smoothing effect is still very prominent.

Figure 1.1 shows some common problems (or artifacts) introduced by the image interpolation. Chapter 2 will address these artifacts in detail.

#### Chapter 1 Introduction

The wavelet transform has proven effective on image compression. For example, JPEG 2000 uses wavelet transforms for lossy and lossless compression [6]. Also the lifting scheme of the wavelet transform is very suitable for fast computation and hardware implementation [36]. However, image interpolation using wavelet transforms has not proven to be as successful as image compression. Chapter 4 will address the problems in this area.



original image



bilinear (blurred image)



replication (jagged contours)



Li & Orchard (distorted texture)

**Figure 1.1** Some problems caused by the image interpolation (*Note: In the experiment, the original image is reduced by half using an interpolation method; then the same interpolation method is used again to enlarge the reduced image by 2.)* 

A good quality image is non-blurred and artifact-free and has sharp edges. However, the problem of evaluating image quality remains an open question because of the lack of an efficient

Chapter 1 Introduction

metric. The subjective metrics, like the subjective mean opinion score (MOS), give more accurate results, but are often time-consuming, expensive, and inconvenient. Objective metrics are easy to automate and rapid to calculate. Since they are convenient, objective metrics are still an active research problem. The current objective metrics, like mean square error (MSE), peak signal to noise ratio (PSNR), etc., do not accurately reflect the real quality of the image.

# **1.5 Overview and Contribution of this Thesis**

This thesis proposes a new image interpolation method based on wavelet transforms. Through comparisons with traditional and state-of-the-art image interpolation methods, we show that the new method is very promising. Briefly, the contributions of this thesis can be outlined as follows:

- Proposes a new wavelet-based image interpolation algorithm that is simple, fast, and effective
- Discusses the detailed problems of image interpolation in the space and wavelet domains
- Explores a new image quality measurement

The chapter contents of this thesis are as follows:

Chapter 1 introduces the concept, origins, and problems of image interpolation; lists the contributions of this thesis.

Chapter 2 discusses some common artifacts; introduces image quality measurements including MSE, PSNR, SSIM, and IQ; reviews the wavelet transform theory and lifting scheme.

Chapter 3 details two traditional image interpolation methods: bilinear and bicubic; details two state-of-the-art image interpolation algorithms: Kimmel's and Li & Orchard's.

Chapter 4 introduces existing image interpolation methods using wavelet transforms; details the problems and performances of wavelet transform for image interpolation.

Chapter 5 introduces a new image interpolation method based on wavelet transforms; lists experiment results and analyzes the performance of the new method.

Chapter 6 lists conclusions and further research.

# PRELIMINARIES

## 2.1 Introduction

To facilitate an understanding of the contents in this thesis, this chapter reviews some fundamental concepts. In the first part, the concept of image quality and some image quality measurements are introduced since they are good tools for evaluating the performance of interpolation methods. It is worth mentioning that the structural similarity (SSIM) measurement recently proposed is better than traditional measurements such as PSNR and MSE. The last part of this chapter provides a tutorial on wavelet multiresolution theory and lifting scheme. These concepts form the basics of our new image interpolation approach.

# 2.2 Review of Image Quality Measurement

#### 2.2.1 Introduction to Image Quality

What about the performance of an image interpolation method? To answer this question, one must look into the image quality of the interpolated image. There are many factors that influence the quality of an image, such as contrast, sharpness, and brightness. Among these factors, contrast is one of the most important for image quality since human eyes are more sensitive to it. Therefore, most image quality measurements give contrast more weight in an overall assessment of image quality, as seen in the later sections of this chapter. Let  $I_{max}$ ,  $I_{min}$  and  $I_B$  as the maximum, minimum, and background intensities of the image, image contrast can be defined as:

$$C = \frac{I_{\max} - I_{\min}}{I_R} \tag{2.1}$$

Besides these factors, one has to consider artifacts since they often come along with the interpolated image and degrade the image quality. Aliasing, blurring, ringing, and blocking are four common artifacts.

Digital images are susceptible to aliasing. Suppose that the highest frequency is finite and that the function is band-limited. The Shannon sampling theorem states that if the function is sampled at a rate equal to or greater than twice its highest frequency, it is possible to recover completely the original function from its samples. In practice, it is impossible to meet the key condition of the sampling theorem for infinite highest frequency or non band-limited function. This limitation causes additional frequency components to be introduced into the sampled function and corrupted image. One way to reduce the aliasing effects on an image is to use a low pass filter to filter out part of the high-frequency components by blurring the image prior to sampling. The effect of aliased frequencies can be seen under the right conditions in the form of so-called Moiré patterns [3]. Figure 2.1 illustrates aliasing.



Figure 2.1 Aliasing (a) Original image; (b) image with aliasing distortion in the centre; (c) image with less aliasing.

Blurring is caused by the lose of many high-frequency components. The resulting image appears to be out of focus. Images interpolated by a low-order interpolant like bilinear interpolation are more easily blurred than ones interpolated by a high-order interpolant like bicubic interpolation. Magnified images tend to be more blurred. Figure 2.2 shows two images with different blurring effects.

Ringing arises because most good synthesis functions are oscillating waves, such as wavelet functions and discrete cosine functions. Ringing often happens around steep edges known as the Gibbs phenomenon [1] where limited numbers of oscillating functions are used to approximate a steep edge. An appropriate function or high sampling rate can reduce ringing effects. Figure 2.3 shows ringing effects.

Blocking (zigzag) arises when the support of the interpolant is finite. The best example is the nearest neighbor interpolation. Blocking will become worse after the image is magnified several times with this method. Blocking is often easy to see around steep edges. Block transforms, like discrete cosine transform (DCT), bring another type of blocking. This type of

blocking is easy to see in high compressed JPEG images. Figure 2.4 presents a typical case of blocking.







Figure 2.3 Ringing (a) Original image; (b) approximated image with ringing (sharp changes in intensity around edges).



Figure 2.4 Blocking (a) Original image; (b) Low quality image with zigzag edges; (c) High quality image with smooth edges.

There are two kinds of image quality measurements: objective and subjective metrics. Since all images are ultimately viewed by people, subjective metrics are more precise than objective ones. The subjective mean opinion score (MOS) is a popular method. In practice, however, subjective evaluation is usually too inconvenient, time-consuming, and expensive. Therefore, a great deal of effort has been made in recent years to develop objective image quality metrics that correlate with perceived quality measurement. Unfortunately, only limited success has been achieved [9]. Many people are thus still using traditional approaches, such as mean squared error (MSE) and peak signal-to-noise ratio (PSNR). Objective image quality metrics can further be classified according to the availability of the reference image. Most approaches are known as full-reference, meaning that a complete reference image is known. However, in many practical applications, the reference image is not available, and a blind quality assessment approach has to be performed.

#### 2.2.2 Full-Reference Image Quality Measurement

#### 2.2.2.1 Mean Squared Error (MSE)

Let f(x, y) represent an input image (reference image) and let  $\hat{f}(x, y)$  denote an estimate or approximation of f(x, y), the error e(x, y) between f(x, y) and  $\hat{f}(x, y)$  can be defined as

$$e(x, y) = \hat{f}(x, y) - f(x, y)$$
(2.2)

Therefore, the total error between the two images is

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x,y) - f(x,y)]$$

where the images are of size  $M \times N$ . The mean squared error between f(x, y) and  $\hat{f}(x, y)$  is then the squared error averaged over the  $M \times N$  array, or

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2$$
(2.3)

In this thesis, the root-mean-square error is used instead of the mean squared error, that is,

$$RMSE = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2\right]^{1/2}$$
(2.4)

#### 2.2.2.2 Peak Signal-to-Noise Ratio (PSNR)

One problem with mean squared error is that it depends strongly on the image intensity range. An MSE of 100.0 for an 8-bit image looks dreadful; but an MSE of 100.0 for a 10-bit image is barely noticeable.

Peak Signal-to-Noise Ratio (PSNR) avoids this problem by scaling the MSE according to the image intensity range:

$$PSNR = -10\log_{10}\frac{MSE}{S^2}$$
(2.5)

where S is the maximum pixel value. For an 8-bit image, S is equal to 255. PSNR is measured in decibels (dB). The PSNR measure is also not ideal but is in common use. Its main failing is that the signal strength is estimated as  $S^2$ , rather than the actual signal strength for the image. PSNR is a good measure for comparing restoration results for the same image.

#### 2.2.2.3 Structural Similarity Index (SSIM)

MSE and PSNR are simple to calculate and have clear physical meanings. But they are not very well matched to perceived visual quality. Therefore, new image quality measurements often take advantage of the known characteristics of the human visual system (HVS). Structural SIMilarity (SSIM) based image quality assessment is one such successful approaches. It follows that a measure of structural information change can provide a good approximation to perceived image distortion. Wang et al. [12] recently demonstrated its promise through a set of intuitive examples, as well as a comparison to both subjective ratings and state-of-the-art objective methods on a database of images compressed with JPEG and JPEG2000. Figure 2.5 shows the SSIM measurement system.

For an image f(x, y), the mean intensity is

$$\mu_f = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$
(2.6)

and the standard deviation is given by

$$\sigma_{f} = \left[\frac{1}{(M-1)(N-1)} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x,y) - \mu_{f}]^{2}\right]^{1/2}$$
(2.7)

The correlation of images  $f_1(x, y)$  and  $f_2(x, y)$  is defined as:

$$\sigma_{f_1 f_2} = \frac{1}{M - 1} \frac{1}{N - 1} \sum_{x=0}^{M - 1} \sum_{y=0}^{N - 1} [f_1(x, y) - \mu_{f_1}] [f_2(x, y) - \mu_{f_2}]$$
(2.8)



Figure 2.5 Diagram of the structural similarity (SSIM) measurement system.

Suppose  $C_1$ ,  $C_2$ , and  $C_3$  are constants, for luminance comparison, Wang et al. define

$$l(f_1, f_2) = \frac{2\mu_{f_1}\mu_{f_2} + C_1}{\mu_{f_1}^2 + \mu_{f_2}^2 + C_1}$$
(2.9)

The contrast comparison function takes a similar form:

$$c(f_1, f_2) = \frac{2\sigma_{f_1}\sigma_{f_2} + C_2}{\sigma_{f_1}^2 + \sigma_{f_2}^2 + C_2}$$
(2.10)

And the structure comparison function is given by

$$s(f_1, f_2) = \frac{\sigma_{f_1 f_2} + C_3}{\sigma_{f_1} \sigma_{f_2} + C_3}$$
(2.11)

Finally, the SSIM index between images  $f_1(x, y)$  and  $f_2(x, y)$  is the combination of the three comparisons:

$$SSIM(f_1, f_2) = \left[ l(f_1, f_2) \right]^{\alpha} \left[ c(f_1, f_2) \right]^{\beta} \left[ s(f_1, f_2) \right]^{\gamma}$$
(2.12)

where  $\alpha > 0$ ,  $\beta > 0$ , and  $\gamma > 0$  are parameters used to adjust the relative importance of the three components.

In practice, the local statistics  $\mu$  and  $\sigma$  are computed within a local 8×8 square window, and a mean SSIM (MSSIM) index is used to evaluate the overall image quality. In addition to

13

MSSIM, we can get a quality map. To reduce undesirable blocking artifacts in the map, a circular-symmetric Gaussian weight function is introduced to modify the local statistics. Figure 2.6 shows an example of a quality map. In the quality map, white areas represent good quality, and dark areas represent bad quality.



Figure 2.6 Illustration of the quality map from the MSSIM measurement (a) Original image. (b) Restored image. (c) Quality map of restored image.

#### 2.2.3 Blind Image Quality Measurement

Blind image quality measurement refers to the problem of evaluating the visual quality of an image without any reference. There are several reasons for the need for blind image quality assessment: (1) No reference image exists; (2) The emphasis is more on quality instead of fidelity; (3) The reference image and the interpolated image have different sizes or depths. Though blind image quality measurement is desirable in many areas, there are only a few existing approaches. Nill et al. proposed an IQM approach derived from the digital image power spectrum [10]. The measure incorporates a representation of the human visual system, a novel approach to account for directional differences in perspective (scale) for obliquely acquired scenes, and a filter developed to account for imaging system noise specifically evidenced in the image power spectra. The authors demonstrate a very good correlation between this objective quality measure and visual quality assessments through experiments.

The two-dimensional discrete Fourier transform (DFT) of an image f(x, y) of size  $M \times N$  is given by the equation

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)}$$
(2.13)

The power spectrum is the square of the magnitude of the Fourier transform of the image:

$$P(u,v) = R^{2}(u,v) + I^{2}(u,v)$$
(2.14)

15

where R(u, v) and I(u, v) are the real and imaginary parts of F(u, v), respectively.

It is common practice to multiply the input image by  $(-1)^{x+y}$  before computing the Fourier transform so that the DC component (zero frequency) is shifted to the center of the power spectrum. Figure 2.7 illustrates the power spectrum of images.



Figure 2.7 Power spectrum (a) Original image; (b) and (c) Distorted images.

The power spectrum contains information on the sharpness, contrast, and detail of the image, and these are all components of image quality. The power spectrum H is normalized by its DC power (zero frequency power), resulting in image contrast having a major impact on the computed image quality. The polar representation of the power spectrum is given by

$$P(\rho,\theta) = \frac{|H(\rho,\theta)|^2}{|H(0,0)|^2 / MN}$$
(2.15)

The IQM is derived from the normalized 2-D power spectrum  $P(\rho, \theta)$  weighted by the square of the modulation transfer function (MTF) of the human visual system  $A^2(T\rho)$ , the directional scale of the input image  $S(\theta_1)$ , and the modified Wiener noise filter  $W(\rho)$ :

$$IQM = \frac{1}{MN} \sum_{\theta = -180^{\circ}}^{180^{\circ}} \sum_{\rho = 0.01}^{0.5} S(\theta_1) W(\rho) A^2(T\rho) P(\rho, \theta)$$
(2.16)

## 2.3 Review of Wavelet Transforms

#### 2.3.1 Background

In recent years, wavelets have been widely used in image processing. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small waves, called wavelets, of varying frequencies and limited durations. These characteristics allow them to provide the equivalent of a musical score for an image, revealing not only what notes (or frequencies) to play but also when to play them. Conventional Fourier transforms, on the other hand, provide only the notes or frequency information; temporal information is lost in the transformation process.

Although the first wavelet, Haar transform, can be traced back to 1909, wavelets were first shown to be powerful tools only in the 1980s. In 1987, Stephane Mallat introduced multiresolution theory [1], a big step in wavelets research. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including subband coding from signal processing [4], quadrature mirror filtering from digital speech recognition, and pyramidal image processing [2]. It allowed researchers to construct their own family of wavelets using the criteria of multiresolution theory. Around 1988, Ingrid Daubechies used the idea of multiresolution analysis to construct Daubechies wavelets with compact support and orthogonality, and Daubechies wavelets have become the cornerstone of wavelet applications today.

In 1992, Cohen, Daubechies, and Feauveau established the theory of biorthogonal wavelet systems [1]. This family of wavelets exhibits the property of linear phase, which is needed for signal and image reconstruction. Unlike the Daubechies wavelet system which has a single wavelet, the biorthogonal wavelet system uses two wavelets, one for decomposition and another for reconstruction. Thus interesting properties are derived. It is possible to construct smooth biorthogonal wavelets of compact support that are either symmetric or antisymmetric. This property is impossible for orthogonal wavelets except in the trivial case — Haar wavelet. This symmetry can offer significant benefits in many applications such as image compression.

In 1995, Sweldens proposed the lifting scheme [36], a technique used to improve wavelet properties and construct wavelet basis over non-translation invariant domains. It also led to fast polyphase implementations of filter bank decompositions. Then in 1996, Calderbank, Daubechies, Sweldens, and Yeo proposed a systematic lifting-based technique for constructing reversible versions of any 2-band wavelet transform [34].

16

Besides Haar, Daubechies, and Biorthogonal wavelets, there are many other families of wavelets such as Coiflets, Symlets, and Morlet. Figure 3.1 shows several different families of wavelets.



Figure 2.8 Several different families of wavelets.

Wavelet transforms have proven to be very efficient and effective in analyzing a very wide class of signals and phenomena. The main reasons are (1) the number of the wavelet coefficients drops off rapidly with the increase of the scale of the decomposition for a large class of signals. This is why wavelets are so effective in signal and image compression, denoising, and detection; (2) the wavelet transform allows a more accurate local description and separation of signal characteristics in both time and frequency domains; (3) wavelets are adjustable and adaptable to fit individual applications; (4) The generation of wavelets and the calculation of the discrete wavelet transform is well matched to digital computers.

17

#### 2.3.2 Multiresolution Analysis

In multiresolution analysis (MRA), a scaling function  $\varphi(x)$  is used to create a series of approximations of a function or an image, each differing by a factor of 2 from its nearest neighbouring approximations. A wavelet function  $\psi(x)$  is used to encode the difference in information between adjacent approximations.

For all  $i, k \in \mathbb{Z}$  and  $\varphi(x) \in L^2(\mathbb{R})$ , we define a set of scaling functions

$$\varphi_{ik}(x) = 2^{j/2} \varphi(2^j x - k) \tag{2.17}$$

Here, the location k determines the position of  $\varphi_{j,k}(x)$  along the x-axis, the scale j determines  $\varphi_{j,k}(x)$ 's width, and  $2^{j/2}$  controls its height. We denote the V subspace spanned over k for any j as

$$V_j = \overline{Span\{\varphi_{j,k}(x)\}}$$



Figure 2.9 Scaling and wavelet function spaces

Similarly, we define a set of wavelets

$$\psi_{i,k}(x) = 2^{j/2} \psi(2^j x - k) \tag{2.18}$$

and the W subspace spanned over k for any j as

$$W_j = \overline{Span\{\psi_{j,k}(x)\}}$$

Since wavelet spaces reside within the spaces spanned by the next higher resolution scaling functions, we can now express the space of all measurable, square-integrable functions as

$$L^{2}(R) = V_{j_{0}} \oplus W_{j_{0}} \oplus W_{j_{0}+1} \oplus \dots$$

where  $j_0$  is an arbitrary starting scale. Therefore, any function  $f(x) \in L^2(R)$  can be expanded as a wavelet series:

$$f(x) = \sum_{k} c_{j_0,k} \varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty} \sum_{k} d_{j,k} \psi_{j,k}(x)$$
(2.19)

where  $c_{j_0,k}$  and  $d_{j,k}$  are called the approximation coefficients and the detail coefficients, respectively.

Consider the relationships between the discrete wavelet transform (DWT) coefficients of adjacent scales:

$$c_{j,k} = \sum_{n} c_{j+1,n} h_0(n-2k)$$
(2.20)

$$d_{j,k} = \sum_{n} d_{j+1,n} h_1(n-2k)$$
(2.21)

where,  $h_0$  is a low-pass filter (scaling filter) and  $h_1$  is a band-pass filter (wavelet filter). It is equivalent that the approximation or detail coefficients in the scale j+1 become ones in the scale j after passing a low-pass or band-pass filter and a down sampler. Therefore, we can use the analysis filter bank [33] in Figure 2.10 to compute DWT coefficients at two or more successive scales. This is called the fast wavelet transform (FWT). Similarly, the synthesis filter bank is used to reconstruct the function (compute the inverse wavelet transform). For the orthogonal wavelet, analysis and synthesis filter banks use the same pair of filters. And for the biorthogonal wavelet, there are two pairs of filters, one for analysis and another for synthesis. Figure 2.13~2.22 shows the filters, the scaling and wavelet functions of several biorthogonal wavelets.



Figure 2.10 A two-stage or two-scale FWT analysis bank and its frequency-splitting characteristics.

For two-dimensional functions as in images, a two-dimensional scaling function and three two-dimensional wavelets are required. Each is the product of a one-dimensional scaling function and a corresponding wavelet (assume the wavelets are separable).

$$\varphi(x, y) = \varphi(x)\varphi(y) \tag{2.22}$$

$$\psi_H(x, y) = \psi(x)\varphi(y) \tag{2.23}$$

$$\psi_{\nu}(x, y) = \varphi(x)\psi(y) \tag{2.24}$$

$$\psi_{\mathcal{D}}(x, y) = \psi(x)\psi(y) \tag{2.25}$$

These wavelets measure intensity variations for images along different directions:  $\psi_H$  measures variations along columns,  $\psi_V$  responds to variations along rows, and  $\psi_D$  corresponds to variations along diagonals. The two-dimensional wavelet transform will produce a set of approximation coefficients and three sets of detail coefficients – the horizontal, vertical, and diagonal details. Similar to the one-dimensional fast wavelet transform, the two-dimensional fast wavelet transform can be implemented by the filter bank shown in Figure 2.11

20



Figure 2.11 The two-dimensional fast wavelet transform: (a) the analysis filter bank; (b) the synthesis filter bank; and (c) the resulting decomposition.

#### 2.3.3 Lifting Scheme

The lifting scheme, a simple construction of second-generation wavelets, uses a simple relationship between all multiresolution analyses with the same scaling function. It can be used to custom design a wavelet for a particular application. Such wavelets can be adapted to intervals, domains, surfaces, weights, and irregular samples. The lifting scheme also leads to a faster, in-place calculation of the wavelet transform [36].

From the transform point of view, we describe the lifting scheme as a lazy wavelet transform [3] followed by alternating lifting (predict) steps and dual lifting (update) steps. Figure 2.12 shows the forward and inverse wavelet transforms using lifting.





The forward lifting algorithm can be implemented in three stages:

(1) Split The original signal x(n) is split into its even and odd indexed samples (lazy wavelet transform):

$$s_0(l) = x(2l)$$
 (2.26)

$$d_0(l) = x(2l+1) \tag{2.27}$$

(2) Predict and Update This stage is executed as m sub-steps, where the odd and even parts are filtered by the prediction and update filters.

• A predict step consists of applying a predict filter to the even samples and subtracting the result from the odd samples:

$$d_{i}(l) = d_{i-1}(l) - \sum_{k} p_{i}(k)s_{i-1}(l-k)$$
(2.28)

• An update step consists of applying an update filter to the odd samples and subtracting the result from the even samples:

$$s_i(l) = s_{i-1}(l) - \sum_k u_i(k) d_i(l-k)$$
(2.29)

#### (3) Normalize:

$$s(l) = s_m(l) / K$$
, approximation coefficients (2.30)

$$d(l) = Kd_m(l)$$
, detail coefficients (2.31)

As always, we can find the inverse transform by reversing the operations and flipping the signs. And the normalize step is often omitted to reduce the computational time.

Wavelets with finite filters can always be factored into lifting steps [35]. Using lifting, one can build invertible wavelet transforms that map integers to integers [34]. Table 2.1 lists some wavelet transforms using the lifting scheme.

- 5/3 symmetric biorthogonal transform [34, 6]
- 9/7-M symmetric biorthogonal transform [34, 6]
- 5/11-C symmetric biorthogonal transform [34, 6]
- 2/10 anti-symmetric biorthogonal transform [6]
- 9/7-F symmetric biorthogonal transform [34, 6]

For each wavelet transform, Table 2.2 lists some parameters, such as vanishing moment, regularity, and filter length. These parameters directly or indirectly influence the performance of the wavelet transform [37]. Table 2.3 lists the computational complexity of each wavelet transform. Note that the set number in parentheses corresponds to the situation that all multiplications are converted to shift and add operations.

Table 2.1 Sou	me wavelet trans	forms using t	the lifting scheme
---------------	------------------	---------------	--------------------

Name	Wavelet Transform	Normalization Factor
5/3	$d(l) = x(2l+1) - \frac{x(2l) + x(2l+2)}{2}$	$K = \frac{\sqrt{2}}{2}$
	$s(l) = x(2l) + \frac{d(l-1) + d(l)}{4}$	
9/7-M	$d(l) = x(2l+1) - \frac{x(2l) + x(2l+2)}{16/9} + \frac{x(2l-2) + x(2l+4)}{16}$	$K = \frac{\sqrt{2}}{2}$
	$s(l) = x(2l) + \frac{d(l-1) + d(l)}{4}$	
5/11-C	$d_1(l) = x(2l+1) - \frac{x(2l) + x(2l+2)}{2}$	$K = \frac{\sqrt{2}}{2}$
	$s(l) = x(2l) + \frac{d_1(l-1) + d_1(l)}{4}$	
	$d(l) = d_1(l) - \frac{-s(l-1) + s(l) + s(l+1) - s(l+2)}{16}$	
2/10	$d_1(l) = x(2l+1) - x(2l)$	$K = \frac{\sqrt{2}}{2}$
	$s(l) = x(2l) + d_1(l)/2$	-
	$d(l) = d_1(l) - \frac{22(s(l+1) - s(l-1)) + 3(s(l-2) - s(l+2))}{64}$	
9/7-F	$d_1(l) = x(2l+1) + \alpha(x(2l) + x(2l+2))$	<i>K</i> ≈1.149604398
	$s_1(l) = x(2l) + \beta(d_1(l-1) + d_1(l))$	$\alpha \approx -1.586134342$
	$d(l) = d_1(l) + \gamma(s_1(l) + s_1(l+1))$	$\beta \approx -0.052980118$
	$s(l) = s(l) + \delta(d(l) + d(l-1))$	$\gamma \approx 0.8829110762$
		$\delta \approx 0.4435068522$

Name	N <sub>r</sub>	N <sub>d</sub>	R <sub>r</sub>	R <sub>d</sub>	L <sub>r</sub>	$L_d$
5/3	2	2	0.000	1.000	3	5
9/7-M	4	2	0.142	2.000	7	•9
5/11-C	4	2	0.000	2.142	11	5
2/10	5	1	0.000	1.804	10	2
9/7-F	4	4	1.034	1.701	7	9

Table 2.2 Parameters	of	some	wave	let	transforms
----------------------	----	------	------	-----	------------

**Parameter Definition:** 

•  $N_d$ : number of vanishing moments of analyzing wavelet

•  $N_r$ : number of vanishing moments of synthesizing wavelet

- $R_d$ : regularity of analyzing scaling function [6]
- $R_r$ : regularity of synthesizing scaling function [6]
- $L_r$ : The low-pass filter length in the decomposition
- $L_d$ : The low-pass filter length in the reconstruction

 Table 2.3 Computational complexity of some wavelet transforms [6]

Name	additions	shifts	multiplies	totals
5/3	5	2	0	7
9/7-M	8(9)	2(3)	1(0)	11(12)
5/11-C	10	. 3	0	13
2/10	7(10)	2(6)	2(0)	11(16)
9/7-F	12(26)	4(18)	4(0)	20(44)







Figure 2.14 9/7-M filters







Figure 2.16 2/10 filters



Figure 2.18 5/3 scaling and wavelet functions

0.5

0

-0.5

0





0.5

-0.5

0

10




Figure 2.22 9/7-F scaling and wavelet functions

# STATE OF THE ART IN IMAGE INTERPOLATION

# 3.1 Introduction

The interpolation methods all work in a fundamentally similar way. In each case, to determine the value of an interpolated pixel, you find the point in the input image that the output pixel corresponds to. You then assign a value to the output pixel by computing a weighted average of some set of pixels in the vicinity of the point. For conventional approaches, the weightings are based on the distance each pixel is from the point. The methods differ in the set of pixels that are considered. For edge-directed approaches, the weightings are related to the direction of the local edge. The following sections will detail these approaches.

## 3.2 Conventional Approaches

Conventional image interpolation approaches are actually driven from polynomial approximation functions in one dimension. For example, the bilinear interpolation is from the linear interpolation, and the bicubic interpolation is from the cubic interpolation. The linear interpolation kernel is a low-order polynomial function, and the cubic interpolation kernel is a high-order polynomial function. The higher the order of the polynomial, the more neighbors will be considered. Therefore, for bilinear interpolation, the output pixel value is a weighted average of pixels in the nearest 2-by-2 neighborhood; for bicubic interpolation, the output pixel value is a weighted average of pixels in the nearest 4-by-4 neighborhood. The number of pixels considered affects the complexity of the computation. Therefore the bicubic method takes longer than the bilinear. However, the greater the number of pixels considered, the more accurate the effect is. Therefore, there is a trade-off between processing time and quality. In this thesis, we use bilinear and bicubic algorithms from the matlab function *imresize*.

## 3.2.1 Bilinear Interpolation

As we know, bilinear interpolation comes from linear interpolation, which is a first-order polynomial approximation method. Its concept is very simple, and its 1D interpolation kernel is

$$u(s) = \begin{cases} 1 - |s|, & |s| < 1\\ 0, & elsewhere \end{cases}$$
(3.1)

where  $s = (x - x_k)/h$  and h is the sampling increment. In fact, this is a triangle or hat function that has C<sup>0</sup> regularity (continuous but not differentiable). The triangular function u(s) corresponds to a modest low-pass filter in the frequency domain (Fig 3.1). When  $x_k < x < x_{k+1}$ , the interpolated value of x is

$$g(x) = (1-s)f(x_k) + sf(x_{k+1})$$
(3.2)



Figure 3.1 Linear interpolation (a) kernel; (b) Magnitude of Fourier transform.



Figure 3.2 Illustration of bilinear interpolation

#### CHAPTER 3 State of the Art in Image Interpolation

Bilinear interpolation is an interpolation method that uses the values from the four surrounding pixels, that is, above, below, right, and left of the spot where the new pixel is to be created. The new pixel value is determined by calculating the weighted average of the four closest pixels (a 2x2 array) based on distance. Here an example shows how to use the bilinear interpolation in practice. Figure 3.2 is an image with input pixels '00', '01', '10' and '11'. The intention is to calculate the intensity of the point 'z'. First, we calculate the intensity of points 'a' and 'b' by using the linear interpolation ( $0 < \Delta y < 1$ ).

$$I_a = I_{00} + (I_{01} - I_{00})\Delta y \tag{3.3}$$

$$I_b = I_{10} + (I_{11} - I_{10})\Delta y \tag{3.4}$$

Therefore, the intensity of the point 'z' is the linear combination of the intensities of points 'a' and 'b'  $(0 < \Delta x < 1)$ :

$$I_z = I_a + (I_b - I_a)\Delta x \tag{3.5}$$

The low computation expense of bilinear interpolation makes its use popular. But it tends to make an image softer; that is, the contrast is reduced because of averaging neighboring values.

#### **3.2.2 Bicubic Interpolation**

Since the low-order linear interpolation cannot approximate the given points accurately, the piecewise three-order polynomials are used to make the interpolation more effective. There are many kinds of cubic interpolation methods. In this thesis, Keys' interpolation kernel [14] is used:

$$u(s) = \begin{cases} \frac{3}{2}|s|^{3} - \frac{5}{2}|s|^{2} + 1, & 0 \le |s| < 1 \\ -\frac{1}{2}|s|^{3} + \frac{5}{2}|s|^{2} - 4|s| + 2, & 1 \le |s| < 2 \\ 0, & 2 \le |s| \end{cases}$$
(3.6)

where  $s = (x - x_k)/h$  and h is the sampling increment. This kernel is smoother than that of linear interpolation with C<sup>1</sup> regularity. Sometimes, it is called cubic convolution interpolation kernel (Figure 3.3). Keys also gave a simple algorithm for computing the interpolated value when considering boundary conditions. Suppose  $x_k < x < x_{k+1}$ , the cubic convolution interpolation function is



Figure 3.3 Cubic interpolation (a) kernel; (b) Magnitude of Fourier transform.

$$g(x) = c_{k-1}(-s^3 + 2s^2 - s)/2 + c_k(3s^3 - 5s^2 + 2)/2 + c_{k+1}(-3s^3 + 4s^2 + s)/2 + c_{k+2}(s^3 - s^2)/2$$
(3.7)

where  $s = (x - x_k)/h$  and  $c_k = f(x_k)$  for k=0, 1, 2,..., N;  $c_{k-1} = 3f(x_0) - 3f(x_1) + f(x_2)$ and  $c_{N+1} = 3f(x_N) - 3f(x_{N-1}) + f(x_{N-2})$ .

Bicubic interpolation, a 2-D extension of cubic interpolation, determines the values of new pixels by calculating the weighted average of the closest 16 pixels (a 4x4 array) based on distance. Figure 3.4 shows that the intensities of pixels 'a', 'b', 'c', and 'd' are calculated first using the equation (3.7):



Figure 3.4 Illustration of bicubic interpolation

$$\frac{\text{CHAPTER 3} = \text{State of the Art in Image Interpolation}}{I_a = I_{00}(-\Delta y^3 + \Delta y^2 - \Delta y)/2 + I_{01}(3\Delta y^3 - 5\Delta y^2 + 2)/2}$$
(3.8)  

$$\frac{I_a = I_{00}(-\Delta y^3 + \Delta y^2 - \Delta y)/2 + I_{03}(\Delta y^3 - \Delta y^2)/2}{I_b = I_{10}(-\Delta y^3 + \Delta y^2 - \Delta y)/2 + I_{11}(3\Delta y^3 - 5\Delta y^2 + 2)/2}$$
(3.9)  

$$\frac{I_b = I_{10}(-\Delta y^3 + \Delta y^2 - \Delta y)/2 + I_{13}(\Delta y^3 - \Delta y^2)/2}{I_c = I_{20}(-\Delta y^3 + \Delta y^2 - \Delta y)/2 + I_{21}(3\Delta y^3 - 5\Delta y^2 + 2)/2}$$
(3.10)

$$I_{d} = I_{30} (-\Delta y^{3} + \Delta y^{2} - \Delta y)/2 + I_{31} (3\Delta y^{3} - 5\Delta y^{2} + 2)/2 + I_{32} (-3\Delta y^{3} + 4\Delta y^{2} + \Delta y)/2 + I_{33} (\Delta y^{3} - \Delta y^{2})/2$$
(3.11)

where  $0 < \Delta y < 1$ . The intensity of pixel 'z' is the cubic interpolation of pixels 'a', 'b', 'c', and 'd'  $(0 < \Delta x < 1)$ :

$$I_{z} = I_{a}(-\Delta x^{3} + \Delta x^{2} - \Delta x)/2 + I_{b}(3\Delta x^{3} - 5\Delta x^{2} + 2)/2 + I_{c}(-3\Delta x^{3} + 4\Delta x^{2} + \Delta x)/2 + I_{d}(\Delta x^{3} - \Delta x^{2})/2$$
(3.12)

Bicubic interpolation tends to make an image less blurry than bilinear interpolation. Although this method requires more computational time, it is the default image-enlargement technique in just about every image manipulation program.

## 3.3 Edge-Directed Interpolation

The idea behind Edge-directed interpolation (EDI) is to make sure that the image is not smooth in the directions perpendicular to edges but smooth parallel to edges. This idea coincides with the properties of human perception [19, 20]. The first desirable EDI algorithm was proposed by Allebach and Wong [21]. The Allebach method assumes knowledge of the low-pass filtering kernel (the sensor model) and modifies the edge-directed interpolation so that the filtered high-resolution image fits into the low-resolution image through iterations. A newer method, from Li and Orchard [23, 24], gets rid of several of the disadvantages of Allebach's approach, such as using an edge map, the knowledge of the low-pass filter, and iterations. The approach by Li and Orchard is ideal for strong edges. This approach is computationally expensive and inadequate for the textured areas. Kimmel [22] proposed an edge-directed interpolation method with a low computation expense. It is more practical than the others, though it still blurs the image a little bit.

The detailed algorithms of Kimmel's and Li & Orchard's approaches are in section 3.31 and section 3.3.2.

## 3.3.1 Kimmel's Approach



Figure 3.5 Pixel interpolation in Kimmel's approach (a) given the corner pixels (b) given the side pixels

Ron Kimmel proposed a new image interpolation method for color CCD images in 1999 [22]. In this thesis, Kimmel's approach is used for gray images. As we know, the gradient magnitude can be used as an edge indicator, and its direction can approximate the edge direction. The directional derivatives are approximated at each pixel, based on its eight nearest neighbors on the grid. The gradients at the pixel (i, j) (Figure 3.5), in the x, y, x-diagonal, and y-diagonal directions are defined as follows:

$$D_x(i,j) = \frac{I_{i-1,j} - I_{i+1,j}}{2}$$
(3.13)

$$D_{y}(i,j) = \frac{I_{i,j+1} - I_{i,j-1}}{2}$$
(3.14)

$$D_{xd}(i,j) = \frac{I_{i+1,j+1} - I_{i-1,j-1}}{2\sqrt{2}}$$
(3.15)

$$D_{yd}(i,j) = \frac{I_{i-1,j+1} - I_{i+1,j-1}}{2\sqrt{2}}$$
(3.16)

36

#### CHAPTER 3 State of the Art in Image Interpolation

With the above gradients, the weight of the neighbor in the interpolating process is defined as:

$$W_{i+k,j+l} = \left(1 + D^2(i,j) + D^2(i+k,j+l)\right)^{-1/2}$$
(3.17)

where D is the difference in the direction of (i+k,i+l) and  $k \in \{\pm 1\}, l \in \{\pm 1\}$ . For example,

$$W_{i-1,j-1} = \left(1 + D_{xd}^2(i,j) + D_{xd}^2(i-2,j-2)\right)^{-1/2}$$
(3.18)

Notice that we used  $D_{xd}(i-2, j-2)$  since  $D_{xd}(i-1, j-1)$  is not known.

Therefore, when the corner pixels are known (Figure 3.5 (a)), the intensity value of interpolated pixel (i, j) is

$$I_{i,j} = \frac{I_{i+1,j+1}W_{i+1,j+1} + I_{i+1,j-1}W_{i+1,j-1} + I_{i-1,j+1}W_{i-1,j+1} + I_{i-1,j-1}W_{i-1,j-1}}{W_{i+1,j+1} + W_{i+1,j-1} + W_{i-1,j+1} + W_{i-1,j-1}}$$
(3.19)

and when the side pixels are known (Figure 3.5 (b)), the intensity value of the interpolated pixel (i, j) is

$$I_{i,j} = \frac{I_{i+1,j}W_{i+1,j} + I_{i-1,j}W_{i-1,j} + I_{i,j-1}W_{i,j-1} + I_{i,j+1}W_{i,j+1}}{W_{i+1,j} + W_{i-1,j} + W_{i,j-1} + W_{i,j+1}}$$
(3.20)

Kimmel's method requires two passes: (1) interpolate the pixels that have known values at their corners and (2) interpolate the pixels that have known values at their sides. This type of interpolation smoothes edges and tends to introduce some patchiness around textured areas [48]. This thesis implements Kimmel's algorithm above.

### 3.3.2 Li & Orchard's approach

The basic idea with Li & Orchard's method is that edge directions are invariant to resolution – geometric duality. This means that if we were to actually solve for a curve that modells the edges of some image, we should basically get the same curves for the same image at any resolution. Of course, some small or short edges could be lost in extremely low resolution images. However, actually solving for edge curves is a ridiculously slow process involving partial differential equation (PDE) solvers. Moreover, it only tells us about the image characteristics and not the actual sampling information to draw from the interpolation. However, statistical quantities, and variances in particular will give us information about edges since high local variance quantities means large changes in value, that is, an edge.

Consider Figure 3.6 (a) where the fourth-order linear interpolation is:

$$I_{2i+1,2j+1} = \sum_{k=0}^{1} \sum_{l=0}^{1} W_{2k+l} I_{2(i+k),2(j+l)} = \overline{W}^T \overline{I}$$
(3.21)

where weighting matrix  $\overline{W} = (W_0 \quad W_1 \quad W_2 \quad W_3)^T$  and the intensity matrix of the four nearest neighbors along the diagonal direction  $\overline{I} = (I_{2i,2j} \quad I_{2i+2,2j} \quad I_{2i+2,2j})^T$ .



Figure 3.6 Geometric duality (a) interpolate  $I_{2i+1,2j+1}$  from  $I_{2i,2j}$ ; (b) interpolate  $I_{i,j}(i+j=odd)$  from  $I_{i,j}(i+j=even)$ .

A reasonable assumption made with the natural image source is that it can be modeled as a locally stationary Gaussian process. According to classical Wiener filtering theory, the optimal minimum mean squared error (MMSE) linear interpolation weights are given by

$$\overline{W} = R^{-1}\overline{r} \tag{3.22}$$

where R and  $\vec{r}$  are the local covariances at high resolution. They can be easily estimated from a local window of the low-resolution image using the classical covariance method

$$\hat{R} = \frac{1}{M^2} C^T C , \ \hat{\vec{r}} = \frac{1}{M^2} C^T \vec{y}$$
 (3.23)

where M is the size of the local window,  $\overline{y} = [y_1 \dots y_k \dots y_{M^2}]^T$  is the data vector containing the  $M \times M$  pixels inside the local window, and C is a  $4 \times M^2$  data matrix whose kth column vector is

#### CHAPTER 3 State of the Art in Image Interpolation

the four nearest neighbors of  $y_k$  along the diagonal direction. Figure 3.6 shows the covariances R and  $\bar{r}$  for i + j =even and i + j =odd. Notice that the latter case is just a 45-degree rotation of the former one. Therefore, the weighting matrix is given by

$$\overline{W} = (C^T C)^{-1} (C^T \overline{y}) \tag{3.24}$$



Figure 3.7 An example of Li & Orchard's algorithm.

For example (Figure 3.7), the black-dotted pixels are the pixels from the original image. The pixel (i, j) is interpolated. Let M=2, then

$$\vec{I} = \vec{y} = \begin{pmatrix} I_{11} & I_{21} & I_{12} & I_{22} \end{pmatrix}^{T}$$

$$C = \begin{pmatrix} I_{00} & I_{02} & I_{20} & I_{22} \\ I_{01} & I_{03} & I_{21} & I_{23} \\ I_{10} & I_{12} & I_{30} & I_{32} \\ I_{11} & I_{13} & I_{31} & I_{33} \end{pmatrix}$$

$$\vec{W} = (C^{T}C)^{-1}(C^{T}\vec{y})$$

$$I_{i,j} = \vec{W}^{T}\vec{I}$$

Like Kimmel's method, Li & Orchard's approach requires two passes: (1) interpolate the pixels that have known values at their corners and (2) interpolate the pixels that have known values at their sides. In each pass, when encountering the smooth region (the local variance is

#### CHAPTER 3 State of the Art in Image Interpolation

less than 8), bilinear interpolation is used to replace covariance-based interpolation to avoid the problem of solving the inverse of an ill-conditioned matrix. However, the problem still exists when the local window is small or for some kinds of edges. One good solution is to calculate first the condition number of the covariance R to detect this critical condition [25].

The size of the local window plays an important role in Li & Orchard's approach. A large window can often give stable and reliable results. Meanwhile, the computation is expensive, and the characteristic of the local area is lost. A large window also makes the image smoother and blurs the sharp edges. Therefore, it is better to choose the local window's size adaptively.

Li & Orchard's approach works very well around edges. But it also has some serious problems. One problem is easy to see: a lot of time is needed to compute the inverse matrix. Another problem is that the interpolated image is smoothed in the textured regions. Much effort has gone into solving these problems [25-29].

This thesis implements Li & Orchard's algorithm with a 4×4 local window.

# PROBLEMS OF WAVELET-BASED IMAGE INTERPOLATION

# 4.1 Introduction

A wavelet transform has two noticeable properties:

1) It can concentrate most of the image's energy into its approximation coefficients. For example, when the 'Cameraman' image is decomposed at one scale by the 'bior2.4' wavelet transform, the approximation coefficients contain over 98% of the image's energy.

2) The detail coefficients show the image's edges in the horizontal, vertical and diagonal directions, and then similar patterns appear at different scales (see Figure 4.1).

These two desirable properties make wavelet transforms attractive for use in image interpolation. Since it is easy to obtain reduced images from its wavelet approximation coefficients, we here only discuss wavelet-based image expansion. The principal objective of wavelet-based image expansion is to predict the high-resolution image detail (horizontal, vertical and diagonal coefficients). Here, the approximation coefficients of the high-resolution image are assumed to form the low-resolution image.



**Figure 4.1** Wavelet transform illustration (a) Original image; (b) One-scale forward wavelet transform

There are several detail coefficients predictors in the literature. Huang and Chang's approach uses multilayer perceptron (MLP) from neural networks [40]. Kinebuchi et al use Hidden Markov Trees (HMT) to predict the coefficients at finer scales [41]. Zhu et al propose a statistical estimation scheme [42]. Muresan et al also propose a similar method using the theory of optimal recovery [43]. These methods are often computationally expensive because of the

#### Chapter 4 Problems of Wavelet-based Image Interpolation

training process or because of the use of complex mathematical operations. According to the self-similarity property of wavelet coefficients across scales, Xu et al propose a hybrid approach by combining wavelets with fractals, which is suitable for textured images [44].

However, Mallat's wavelet transform modulus maxima theory [1] received more attention and has been used often in the literature e.g. [46, 47]. This is because of its following property: if a signal is represented by its low-pass approximation coefficients and the modulus maxima (maxima and minima) of high-pass detail coefficients from its wavelet transform, this representation allows an almost perfect reconstruction. Finding a minimum norm signal among those that have the assigned wavelet coefficients at the maxima locations can solve this reconstruction problem. Solving this problem tends to create a signal with modulus maxima at the right locations with the correct values. For discrete signals, this problem is actually an inverse frame problem [1], which can be solved using a conjugate gradient algorithm [1]. Mallat also proposed an alternate projection algorithm [45] that recovers the signal approximations from their wavelet maxima.

Under the wavelet transform modulus maxima theory, a basic wavelet-based interpolation problem is that of predicting the extrema of the detail coefficients of the high-resolution image using the extrema of the detail coefficients of the two or more consecutive low-resolution images. This problem will be detailed in section 4.3. A 2-D separable wavelet transform is generally used for the two-dimension case (image). Most approaches use undecimated symmetric wavelet transform since it can keep the same signs and the same locations of the extrema across scales. Nicolier et al propose a method using decimated wavelet transform [49]. However, for this method, one has to figure out the signs and locations of the extrema. Figure 4.2 is a basic wavelet-based interpolation problem model (one-dimension).

In the following sections, we will detail the pros and cons of wavelet transforms and the methods using the modulus maxima theory.



Figure 4.2 A wavelet-based interpolation problem model

# 4.2 Problems of Wavelet Transforms

### 4.2.1 Approximation Issues

In this section, we discuss the approximation performance of the wavelet transform, and the associated problems when it is used for image interpolation (decompose once, and then reconstruct using approximation coefficients only). As we know, the wavelet transform has excellent approximation capability because it can concentrate most of an image's energy into its approximation coefficients [37]. The comparison of the approximate capabilities of the different wavelet transforms can be done by looking at the quality of the image reconstructed from its approximation coefficients only i.e. without using the detail coefficients. The steps of this procedure are:

- (1) Decompose the test image once using a wavelet transform, and discard its detail coefficients;
- (2) Reconstruct the image using the approximation coefficients only.
- (3) Compare the performance of wavelet transform based interpolations with that of bicubic interpolation. For the latter, bicubic interpolation is used in both the reduction and enlargement processes.

A number of test images (Appendix A) with different characteristics are used. The reference method is the bicubic interpolation. Table 4.1 and Table 4.2 list the overall performance of several wavelet transforms detailed in chapter two. The reason for using these biorthogonal wavelet transforms is their excellent approximation performance [6, 38].

When the bicubic interpolation method is used as a reference, the average gains from different wavelet-based interpolation methods are shown in Table 4.3. According to the data from different measurements, wavelet transform based interpolations are superior to bicubic interpolation with about 1dB increment. This proves that the wavelet transforms have outstanding approximation capabilities. Among these wavelet transforms, the 9/7-F wavelet transform has the best PSNR, and the 9/7-M wavelet transform has the best MMSIM.

For a specific image such as the "cameraman" image, the reconstructed images and quality maps are shown in Appendix B. The reconstructed image using bicubic interpolation has more jaggies at some edges than those obtained using wavelet transform interpolation. If a prefilter is

#### Chapter 4 Problems of Wavelet-based Image Interpolation

used before downsampling, this problem is improved. However, the whole image becomes blurred, and the PSNR is reduced too. It is easy to see that the reconstructed images using wavelet transforms have some artefacts around the edges such as ringing, blurring and blocking. Different wavelet transforms exhibit different results. Overall, the 2/10, 9/7-M, 5/11-C and 9/7-F wavelet transforms are better than others. Anti-symmetric wavelet transforms such as 2/10 have better performance than symmetric ones for some edges such as the high-rise building in the "cameraman" image. However, most edges in the reconstructed image are sensitive to their locations in the image. That is, if the edges are shifted one line up/down or left/right, we will get different results. Some edges at certain locations will be lost. The reason that causes this phenomenon is that the wavelet transform is not shift-invariant. The downsampling operation during decomposition will delete different data if the edges are shifted. Actually, this phenomenon also happens to strong edges. But it is not as noticeable as weak edges in the image. Symmetric wavelet transforms used in this thesis have similar effects due to the shift-variant property. Interested readers could see Nosratinia's paper that details this phenomenon. In his paper, a method is proposed to improve the image quality of the reconstructed image using the shift-variant property of wavelet transforms [51].

Image	Bicubic	5/3	5/11-C	9/7-M	9/7-F	2/10
cameraman	26.34	26.59	26.84	26.85	26.93	26.96
lena	29.79	30.09	30.54	30.56	30.59	30.37
lily	26.71	26.82	27.40	27.41	27.46	27.38
lighthouse	25.15	25.87	25.80	25.84	25.97	25.77
bike	26.31	26.49	26.91	26.93	26.99	26.90
bird	37.05	37.48	38.27	38.28	38.08	37.96
peppers	22.65	22.94	22.99	23.01	23.09	23.10
mandrill	17.01	17.78	17.79	17.82	17.94	17.79
text	13.58	14.63	14.49	14.52	14.64	14.44
text1	32.57	31.75	33.64	33.61	33.49	33.53
chestXR	32.13	33.64	33.88	33.90	33.93	34.33
pelvisXR	31.59	32.28	32.58	32.61	32.71	33.17
brainCT	28.23	28.29	29.43	29.43	29.47	29.71
spineCT	29.52	29.68	30.28	30.29	30.39	30.45
kidneyUS	30.91	31.60	32.13	32.15	32.18	31.96
transUS	31.42	32.33	32.72	32.74	32.71	32.58
boneMR	22.57	23.22	23.28	23.31	23.41	23.26
cesarMR	30.69	31.96	32.37	32.39	32.43	32.37
arteryAng	27.59	26.89	27.25	27.27	27.41	27.55
lungAng	31.55	32.36	32.77	32.78	32.77	32.13

Table 4.1 Performance comparison of wavelet transforms using approximation coefficients (PSNR)

Image	Bicubic	5/3	5/11-C	9/7-M	9/7-F	2/10
cameraman	0.8703	0.8804	0.8831	0.8836	0.8801	0.8849
lena	0.9089	0.9152	0.9196	0.9202	0.9179	0.9180
lily	0.8922	0.8964	0.9067	0.9070	0.9067	0.9059
lighthouse	0.8068	0.8278	0.8267	0.8277	0.8260	0.8242
Bike	0.8739	0.8791	0.8875	0.8880	0.8876	0.8875
Bird	0.9605	0.9640	0.9662	0.9665	0.9614	0.9644
peppers	0.7278	0.7598	0.7588	0.7595	0.7588	0.7576
mandrill	0.6545	0.6935	0.6968	0.6980	0.7019	0.6935
text	0.5436	0.6010	0.5945	0.5966	0.6021	0.6067
text1	0.9707	0.9707	0.9746	0.9743	0.9687	0.9732
chestXR	0.9761	0.9785	0.9795	0.9799	0.9764	0.9785
pelvisXR	0.9514	0.9554	0.9566	0.9574	0.9532	0.9561
brainCT	0.9490	0.9441	0.9543	0.9544	0.9521	0.9555
spineCT	0.9490	0.9495	0.9553	0.9554	0.9531	0.9541
kidneyUS	0.9309	0.9459	0.9522	0.9525	0.9511	0.9511
transUS	0.9292	0.9475	0.9522	0.9522	0.9445	0.9488
boneMR	0.7128	0.7388	0.7396	0.7405	0.7433	0.7405
cesarMR	0.9695	0.9731	0.9780	0.9787	0.9757	0.9780
arteryAng	0.8917	0.8968	0.9017	0.9024	0.9007	0.9017
lungAng	0.9330	0.9324	0.9412	0.9415	0.9390	0.9405

Table 4.2 Performance comparison of wavelet transforms using approximation coefficients (MSSIM)

	-	•	•	<b>c</b> 1.		•	• •	· · · ·	/ C		1 1	- · - ·
'l'abla /	- <b>H</b>	Avaraga	aaine	of wovalat	tranctorme	neina	annrovimation	CONTINIENTS	( rete	rence	nicili	າເຕາ
1 2010 4		AVELASE	2ams	OI WAYCICL	uansionis	using	approximation	coontenents	(ICIC)	renee.	Ulcui	5101
			0			0	11		•			

Image	Bicubic	5/3	5/11-C	9/7-M	9/7-F	2/10
RMSE	·-	-0.83	-1.17	-1.21	-1.31	-1.19
PSNR (dB)	-	+0.47	+0.90	+0.92	+0.96	+0.92
MSSIM	-	+0.0124	+0.0162	+0.0167	+0.0149	+0.0159

Let us look into the local strong edges of the "cameraman" image. Figure 4.3 shows the reconstructed images and values from some measurements. The results show that the 2/10 wavelet transform has outstanding performance. The 5/11-C, 9/7-M and 9/7-F wavelet transforms are slightly better than the bicubic interpolation. The 5/3 wavelet transform is the worst in this case. Ringing and blocking artefacts are clearly seen in these reconstructed images.

From the above discussion, we get the following results: 1) wavelet transform based interpolations are better than bicubic interpolation, and 2) wavelet transform based interpolations need to be improved around edges of interpolated images.



Figure 4.3 Performance comparison of wavelet transforms using approximation coefficients (strong edges)

#### 4.2.2 Magnification Issues

In this section, we examine the magnification problems in the wavelet approaches. Here, the intensity values of the original image are regarded as the wavelet approximation coefficients of the sought-after high-resolution image. The enlarged image is then reconstructed directly from these coefficients. We also look into the performance of different images and look at a local area of the "cameraman" image that has strong edges. Also, the reference method used is the bicubic interpolation method. Since the size of the interpolated image is not the same as that of the original image, the blind image quality IQ measurement along with the contrast are used to compare the performances of the different approaches.

Table 4.4 and Table 4.5 show the performances of different methods when the magnification factor is 2x2. Appendix C also lists the interpolated images for the "cameraman" image. From

#### Chapter 4 Problems of Wavelet-based Image Interpolation

the results, we know that the magnified images using the 9/7-F and 2/10 wavelet transforms have the best IQ and contrast, followed by the ones using bicubic interpolation and the 9/7-M wavelet transform. The 5/11-C and 5/3 wavelet transforms give the worst performance.

If we look into the performance of the local area that has strong edges (Figure 4.4), we find that the magnified images using the 9/7-F wavelet transform have the best IQ. This is followed by the 2/10, 9/7-M and bicubic. The 5/11-C and 5/3 wavelet transforms give the worst performance. From the magnified images shown in Figure 4.4, we can see different artifacts for the different approaches. The 9/7-F and 2/10 yield ringing and blocking artifacts though they have the best contrast. They also introduce some noise in the smooth area. The magnified images using the 5/3 wavelet transform have both blocking and blurring artifacts. The performances of the 9/7-M, 5/11-C and bicubic interpolation methods lie between these two extreme cases.

Based on the above results, we can deduce that: for image enlargement, not all wavelet transform based interpolations are better than the bicubic interpolation though most of them can give images with good contrast. We also deduce that the 9/7-F and 9/7-M wavelet transforms are the best candidates for the purpose of image enlargement.

Image	bicubic	5/3	5/11-C	9/7-M	9/7-F	2/10
cameraman	0.1061	0.0977	0.0988	0.1060	0.1234	0.1263
lena	0.1327	0.1244	0.1255	0.1319	0.1458	0.1466
lily	0.2081	0.1982	0.1996	0.2087	0.2266	0.2256
lighthouse	0.1557	0.1445	0.1466	0.1556	0.1855	0.1959
bike	0.2190	0.1985	0.2012	0.2178	0.2506	0.2518
bird	0.0387	0.0373	0.0375	0.0386	0.0407	0.0401
peppers	0.1204	0.1140	0.1158	0.1230	0.1516	0.1661
mandrill	0.3627	0.3238	0.3307	0.3653	0.4702	0.5112
text	0.3288	0.2977	0.3025	0.3294	0.3863	0.4846
text1	0.0322	0.0298	0.0299	0.0320	0.0351	0.0346
chestXR	0.0604	0.0580	0.0582	0.0599	0.0634	0.0635
pelvisXR	0.0994	0.0956	0.0959	0.0995	0.1071	0.1062
brainCT	0.2335	0.2276	0.2279	0.2346	0.2438	0.2431
spineCT	0.1822	0.1764	0.1770	0.1826	0.1927	0.1922
kidneyUS	0.2225	0.2145	0.2155	0.2224	0.2352	· 0.2337
transUS	0.3428	0.3177	0.3196	0.3344	0.3667	0.3583
boneMR	0.1038	0.0948	0.0963	0.1041	0.1250	0.1334
cesarMR	0.1528	0.1474	0.1476	0.1519	0.1598	0.1614
arteryAng	0.4961	0.4757	0.4762	0.4928	0.5272	0.5339
lungAng	0.0890	0.0825	0.0833	0.0879	0.0970	0.0963
Average	<b>-</b> ,	-0.0115	-0.0101	-0.0004	0.0223	0.0309

Table 4.4 Image enlargement performance of wavelet transforms (IQ)

3

Image	bicubic	5/3	5/11-C	9/7-M	9/7-F	2/10
cameraman	0.5252	0.5177	0.5189	0.5222	0.5279	0.5260
lena	0.5308	0.5233	0.5245	0.5281	0.5338	0.5300
lily	0.5292	0.5221	0.5235	0.5284	0.5358	0.5318
lighthouse	0.3944	0.3857	0.3874	0.3920	0.4020	0.4025
bike	0.5648	0.5518	0.5541	0.5613	0.5727	0.5685
bird	0.3684	0.3657	0.3662	0.3673	0.3687	0.3663
peppers	0.5722	0.5684	0.5695	0.5722	0.5791	0.5784
mandrill	0.5938	0.5783	0.5825	0.5932	0.6178	0.6204
text	0.3018	0.2849	0.2891	0.3008	0.3152	0.3437
text1	0.1707	0.1668	0.1674	0.1700	0.1734	0.1722
chestXR	0.5510	0.5458	0.5460	0.5483	0.5498	0.5468
pelvisXR	0.4571	0.4540	0.4544	0.4570	0.4605	0.4570
brainCT	0.9714	0.9727	0.9732	0.9772	0.9814	0.9789
spineCT	1.0516	1.0430	1.0438	1.0470	1.0499	1.0490
kidneyUS	1.4166	1.3908	1.3921	1.4002	1.4092	1.3955
transUS	1.1730	1.1255	1.1266	1.1402	1.1582	1.1439
boneMR	0.3560	0.3499	0.3515	0.3548	0.3618	0.3621
cesarMR	0.8249	0.8166	0.8169	0.8213	0.8245	0.8194
arteryAng	1.0714	1.0339	1.0369	1.0516	1.0662	1.0628
lungAng	0.5788	0.5710	0.5717	0.5751	0.5795	0.5760
Average	-	-0.0118	-0.0103	-0.0047	0.0032	0.0014

 Table 4.5 Image enlargement performance of wavelet transforms (Contrast)



 $\times 2$ ×4 ×8 Name IQ IQ Contrast Contrast Contrast IQ 0.616649 0.940545 0.300992 0.947662 0.133631 0.951360 5/3 5/11-C 0.617600 0.941616 0.301547 0.949049 0.134025 0.952890 9/7-M 0.627760 0.948154 0.309328 0.958385 0.138666 0.963310 9/7-F 0.647212 0.324151 0.970501 0.147536 0.976823 0.956592 2/10 0.629927 0.310870 0.925046 0.140638 0.923212 0.927282 Bicubic 0.617994 0.931544 0.304503 0.936883 0.137615 0.942432

of wavelet transforms (strong edges)

Figure 4.4 Image enlargement performance

# 4.3 Problems in Approaches Using Mallat's Theory

The modulus maxima (wavelet extrema) in Mallat's theory describe the local maxima or minima of the wavelet coefficients. Mallat pointed out that the decay of the wavelet modulus maximum  $W_j(k)$  across scales is related to the pointwise Lipschitz regularity  $\alpha$  of the signal:

$$|W_j(k)| \le C 2^{-j(\alpha+1/2)}$$
 (4.1)

where, j is the scale of the wavelet transform, and C is a constant. However, wavelet-based image interpolation approaches often assume that the above formula is near-equality at strong edges for some specific wavelets.

Basically, symmetric wavelets are preferred since the non-symmetric wavelets cause incoherent in the signs or locations of the wavelet transform extrema (Table 4.7). Therefore, we here give an example using the symmetric biorthogonal 9/7-F wavelet transform. Figure 4.5 shows the waveforms of the undecimated detail coefficients at scales j=1, 2 and 3 (denoted as S1, S2 and S3 in Figure 4.5) for five different signals. Table 4.6 shows the corresponding numeric values of the detail coefficients around edges. The wavelet extrema are in bold type.



#### Figure 4.5 The propagation of the wavelet extrema

#### Chapter 4 Problems of Wavelet-based Image Interpolation

We can see that the wavelet extrema have good coherences in the signs and locations across scales except for the fifth signal. The problem is to predict the wavelet extrema of the first scale using the second, third and fourth scales. For the first signal, the decay rate of the wavelet extrema is shown in Figure 4.6 (a). Figure 4.6 (b) also shows the decay rate of the second largest wavelet extrema of the third signal. The  $\blacktriangle$  points are the actual values, and the slope of the line is the Lipschitz regularity of the signal. From Figure 4.6, we know that the first situation is under-estimated and the second one is perfect. From Table 4.6, we find that the largest wavelet extrema are often under-estimated. We also notice that some locations have wavelet extrema at scales 2, 3, and 4, but not at the first scale. For this case, the result is often over-estimated. Therefore the assumption of equality of formula 4.1 needs to be adjusted. Otherwise the enlarged image will have more artifacts.

f1	0	0	0	0	0	0	0	127	254	254	254	254	254	254	254
<b>S1</b>	0	0	0	0	8	11	-47	0	47	-11	-8	0	0	0	0
S2	0	0	-3	3	18	-7	-42	0	42	7	-18	-3	3	0	0
<b>S</b> 3	1	-2	-5	10	21	-16	-46	0	46	16	-21	-10	5	2	-1
<b>S4</b>	2	-5	-5	18	23	-24	-54	0	54	24	-23	-18	5	5	-2
-			074	054	054	054	240	010	150	20		0	0	0	
f2	254	254	254	254	254	254	240	210	150	30	0	0	0	0	0
<b>S1</b>	0	0	0	-1	-2	1	-3	7	25	-32	-5	8	2	0	0
<b>S2</b>	0	0	0	-1	-1	-4	-1	18	12	-22	-16	9	7	-1	-1
<b>S</b> 3	0	0	0	0	-2	-7	3	23	11	-24	-21	7	12	0	-3
<b>S4</b>	0	0	0	0	-5	-9	8	30	12	-28	-27	6	17	2	-4
f3	0	0	0	0	0	0	127	254	127	0	0	0	0	0	0
<b>S1</b>	0	0	0	8	11	-55	-11	94	-11	-55	11	8	0	0	0
S2	0	-3	3	21	-10	-60	7	84	7	-60	-10	21	3	-3	0
<b>S</b> 3	-2	-6	12	26	-26	-67	16	92	16	-67	-26	26	12	-6	-2
<b>S4</b>	-6	-6	23	28	-42	-77	24	108	24	-77	-42	28	23	-6	-6
-					0.5	50	100	000	100	50	05		0		
f4	0	0	0	0	25	50	100	200	100	50	25	0	0	0	0
<b>S1</b>	0	2	2	-6	8	-15	-28	73	-28	-15	8	-6	2	2	0
S2	1	2	-1	3	-1	-27	2	44	2	-27	-1	3	-1	2	1
<b>S</b> 3	1	0	2	6	-11	-29	8	45	8	-29	-11	6	2	0	1
<b>S4</b>	0	0	6	7	-20	-33	14	52	14	-33	-20	7	6	0	0

Table 4.6 The wavelet extrema for different signals

## Chapter 4 Problems of Wavelet-based Image Interpolation

6

5.5

5

0

1

f5	0	0	0	0	0	0	0	254	254	254	254	254	254	254	254
<b>S1</b>	0	0	0	0	16	6	-100	100	-6	-16	0	0	0	0	0
S2	1	0	-6	12	25	-38	-46	46	38	-25	-12	6	0	-1	1
<b>S</b> 3	2	-5	-4	24	17	-49	-44	44	49	-17	-24	4	5	-2	2
<b>S4</b>	0	-11	1	34	12	-61	-48	48	61	-12	-34	-1	11	0	0

 Table 4.7 The non-symmetric wavelet extrema

<b>f</b> 1	0	0	0	0	0	0	0	127	254	254	254	254	254	254	254
<b>S1</b>	0	0 .	0	0	0	22	-22	-22	22	0	0	0	0	0	0
<b>S2</b>	0	1	-6	0	32	-1	-52	-1	32	0	-6	1	0	0	0
<b>S</b> 3	5	-11	-15	42	42	-64	-64	42	42	-15	-11	5	1	-1	0





scale

3

4

5

(b)

2

51

.

## 4.4 Summary

From the above analysis, we know that the problem of wavelet-based image interpolation is complicated. One may face several artifacts occurring together such as blurring, ringing and blocking. The best solution is to first choose a good wavelet transform that yields minor artifacts. Then we need to find ways to add detail coefficients so as to reduce the artifacts. Because of the symmetric filters of symmetric wavelets, the transformed coefficients are easy to process. Therefore symmetric wavelets are generally chosen though the anti-symmetric wavelets have good performance for low magnification factors. The 9/7-M and 9/7-F wavelet transforms are good choices overall.

The computational time is usually high for an image interpolation approach using a wavelet transform. One has to use time-consuming methods such as multilayer perceptron (MLP) and Hidden Markov Trees (HMT) to predict the detail coefficients in the high-resolution. If we use Mallat's modulus maxima theory, we still need to know the undecimated detail coefficients at three scales or more. Beside the problem of delivering wrongly estimated wavelet extrema, this approach also needs high computational time to estimate the neighbours of extrema. It also takes more time if long-length wavelet filters are used. However, in order to get good performance, long-length wavelet filters such as the 9/7-F filters are often used. Therefore, we need to find a simple and efficient way to add edge information.

# NEW WAVELET-BASED IMAGE INTERPOLATION APPROACH

# 5.1 Edge Models of Images

Digital images exhibit edges with all kinds of shapes. The step edges with different directions (Figure 5.1) form the fundamental class of edges. In this thesis, the first-order derivatives in the x and y directions (called the x-gradient and the y-gradient) are used to detect these edges.

$$G_{x} = f(x, y) - f(x-1, y)$$
  

$$G_{y} = f(x, y) - f(x, y-1)$$
(5.1)



Figure 5.1 Ideal step edges with different directions

Since the extrema of these gradients determine the main characteristics of edges, we should carefully identify the extrema along the x and y directions. If the current point is the extremum, the following inequalities are valid:

$$|G(i)| > T$$
 and  $|G(i)| \times 1.15 > \max\{G(i-2), G(i-1), G(i+1), G(i+2)\}$  (5.2)

where T is the threshold of extrema, and  $G(i) \in \{G_x(i), G_y(i)\}$ . The value '1.15' is obtained experimentally. The value of the threshold T depends on the content of an image and the wavelet transform. The threshold also influences the computational time. A smaller threshold generally means more detected edges, and more computational time. In this thesis, a value between 50 and 70 is used as the threshold in the 8-bit gray images.

Let us study the characteristics of the images shown in Figure 5.1. Table 5.1 shows the local x- and y- gradients of some of these images having ideal step edges. The extrema of gradients are in bold type. According to the locations of these extrema, the direction of the edge can be determined easily. For example, if the x-gradient and the y-gradient are extrema at a certain point, and the x-gradient of the right neighbor of this point is also an extremum, then there is a 30-degree step edge at this point.

Now, we apply a wavelet transform such as 9/7-M on these ideal step images, and study the characteristics of wavelet approximation coefficients. Here, we call the wavelet approximation coefficients as wavelet-reduced images. We use the same method as above to analyze these images. The results are pretty interesting. When using symmetric wavelets, we get the same extrema patterns of gradients (Table 5.2) like those of the original images though there are more non-zero gradients. When anti-symmetric wavelets are used, we get different and complicated extrema patterns of gradients. For this reason, we use symmetric wavelets to develop the new approach in this thesis. If no specific indication is given, the reported results are all obtained using the 9/7-M and 9/7-F wavelet transforms. Other wavelet transforms may lead to slightly different results.

Let us study what happens if we shift ideal step edges one line down (for 0, 15, 30, 45, 150, and 165 degree edges) or one line right (for 60, 75, 90, 105, 120, and 135 degree edges). We also apply a wavelet transform on these shifted images. The wavelet approximation coefficients (wavelet-reduced images) of these shifted images have the similar extrema patterns of gradients at the same locations as the non-shifted versions. The gradients of wavelet approximation coefficients are differences between the shifted version and the non-shifted version. Here, we call non-shifted version as subtype I, and shifted version as subtype II. By observing the values of the extrema and the nearest neighbors, it is not difficult to distinguish one subtype from the other. For example, a 30-degree edge has two extrema in the x-gradient maps. If the left one is less than the right one, then it belongs to subtype I. Otherwise, it belongs to subtype II. If we further shift ideal step edges down or right, the subtype I and subtype II will appear repeatedly. Table 5.3 lists the extrema patterns of x and y gradients in the wavelet-reduced images from ideal step edges and the rules to distinguish the subtypes.

Table 5.1 Gradients of images having ideal step edges

Gx	Zero-de	egree st	ep edge	Э.		 Gy	Zero-de	egree st	ep edge	•	
0	0	0	0	0	0	0	0	0	0	0	0
192	192	192	192	192	192	0	0	0	0	0	0
0	0	0	0 ·	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
Gx	30-deg	ree step	edge			Gy	30-deg	ree step	edge		
0	0	0	0	192	192	0	0	0	0 ·	192	0
0	0	192	192	0	0	0	0	192	0	0	0
192	192	0	0	0	0	192	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
Gx	45-deg	ree ster	o edge			Gy	45-degree step edge				
0	0	0	0	192	0	0	0	0	0	192	0
0	0	0	192	0	0	0	0	0	192	0	0
0	0	192	0	0	0	0	0	192	0	0	0
0	192	0	0	0	0	0	192	0	0	0	0
Gx	90-deg	ree ster	edge			Gy	90-deg	ree step	edge		
0	0	0	0	0	0	0	0	192	0	0	0
0	0	0	0	0	0	0	0	192	0	0	0
0	0	0	0	0	0	0	0	192	0	0	0
0	0	· 0	0	0	0	0	0	192	0	0	0

Table 5.2 Gradients of wavelet-reduced images having ideal step edges

Gx	Zero-de	egree st	ep edge	e (I)		Gy
24	24	24	24	24	24	
186	186	186	186	186	186	
-24	-24	-24	-24	-24	-24	
3	3	3	3	3	3	
Gx	Zero-de	egree st	ep edge	e (II)		Gy
-24	-24	-24	-24	-24	-24	. 🗌
186	186	186	186	186	186	
24	24	24	24	24	24	
. 3	3	3	3	3	3	
Gx	30-deg	ree step	edge (	I)		Gy
2	7	-23	-1	160	201	
23	-1	160	201	53	-18	-
160	201	53	-18	0	3	1
53	-18	0	3	0	0	-
Gx	30-deg	ree step	edge (	II)		Gy
3	0	-18	52	201	160	
-18	52	201	160	0	-23	-
201	160	0	-23	6	2	1
0	-23	6	2	0	0	-

Gy Zero-degree step edge (1)										
0	0	0	0	0	0					
0	0	0	0	0	0					
0	0	0	0	0	0					
0	0	0	0	0	0.					
Gy	Gy Zero-degree step edge (II)									
<u> </u>		0	<u>^</u>	0	•					

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	. 0
0	0	0	0	0	0

Gy 30-degree step edge (I)

	0				
2	5	-28	27	133	68
-28	27	133	68	-15	-3
133	68	-15	-3	-3	0
-15	-3	3	0	0	0

Зy	30-degree	step	edge	(II)
----	-----------	------	------	------

2	-2	-16	68	133	27
-16	68	133	27	-27	4
133	27	-27	4	2	0
-27	4	2	0	0	0

55

Edge direction	Extrema of gradients	Subtype
	·	I: { $G_x(i-1,j) > G_x(i+1,j)$ and $G_x(i,j) > 0$ } or
0.1	C(i, i) C(i, i+1) C(i, i+2)	$\{G_x(i-1,j) < G_x(i+1,j) \text{ and } G_x(i,j) < 0\}$
0 degree	$G_x(l, j), G_x(l, j+1), G_x(l, j+2)$	II: { $G_x(i-1,j) < G_x(i+1,j)$ and $G_x(i,j) > 0$ } or
		$\{G_{x}(i-1,j) > G_{x}(i+1,j) \text{ and } G_{x}(i,j) < 0\}$
15	$G_x(i, j), G_x(i, j+1), G_x(i, j+2)$	I: $G_x(i, j) < G_x(i, j+2)$
degrees	$G_{y}(i,j)$	II: $G_x(i, j) > G_x(i, j+2)$
30	$G_x(i, j), G_x(i, j+1)$	I: $G_x(i, j) < G_x(i, j+1)$
degrees	$G_{y}(i,j)$	II: $G_x(i, j) > G_x(i, j+1)$
		I: { $G_x(i-1,j) > G_x(i+1,j)$ and $G_x(i,j) > 0$ } or
45	$G_x(i, j), G_x(i-1, j+1), G_x(i-2, j+2)$	$\{G_x(i-1,j) < G_x(i+1,j) \text{ and } G_x(i,j) < 0\}$
degrees	$G_{y}(i, j), G_{y}(i-1, j+1), G_{y}(i-2, j+2)$	II: { $G_x(i-1,j) < G_x(i+1,j)$ and $G_x(i,j) > 0$ } or
		$\{G_x(i-1,j) > G_x(i+1,j) \text{ and } G_x(i,j) < 0\}$
60	$G_x(i,j)$	I: $G_y(i, j) < G_y(i+1, j)$
degrees	$G_{y}(i, j), G_{y}(i+1, j)$	II: $G_y(i, j) > G_y(i+1, j)$
75	$G_x(i,j)$	I: $G_y(i, j) < G_y(i+2, j)$
degrees	$G_{y}(i, j), G_{y}(i+1, j), G_{y}(i+2, j)$	II: $G_y(i, j) > G_y(i+2, j)$
		I: { $G_y(i, j-1) > G_y(i, j+1)$ and $G_y(i, j) > 0$ } or
90	G(i, i) G(i+1, i) G(i+2, i)	$\{G_{y}(i, j-1) < G_{y}(i, j+1) \text{ and } G_{y}(i, j) < 0\}$
degrees	$  \mathbf{G}_{y}(l, j), \mathbf{G}_{y}(l+1, j), \mathbf{G}_{y}(l+2, j)  $	II: { $G_{y}(i, j-1) < G_{y}(i, j+1)$ and $G_{y}(i, j) > 0$ } or
		$\{G_{y}(i, j-1) > G_{y}(i, j+1) \text{ and } G_{y}(i, j) < 0\}$
105	$G_{x}(i,j)$	I: $G_y(i, j+1) < G_y(i+2, j+1)$
degrees	$G_{y}(i, j+1), G_{y}(i+1, j+1), G_{y}(i+2, j+1)$	II: $G_y(i, j+1) > G_y(i+2, j+1)$
120	$G_x(i,j)$	I: $G_y(i, j+1) < G_y(i+1, j+1)$
degrees	$G_{y}(i, j+1), G_{y}(i+1, j+1)$	II: $G_y(i, j+1) > G_y(i+1, j+1)$
·		I: $\{G_{y}(i, j-1) > G_{y}(i, j+1) \text{ and } G_{y}(i, j) > 0\}$ or
135	$G_x(i+1,j), G_x(i+2,j+1), G_x(i+3,j+2)$	$\{G_{y}(i, j-1) < G_{y}(i, j+1) \text{ and } G_{y}(i, j) < 0\}$
degrees	$G_{y}(i, j), G_{y}(i+1, j+1), G_{y}(i+2, j+2)$	II: { $G_{y}(i, j-1) < G_{y}(i, j+1)$ and $G_{y}(i, j) > 0$ } or
		$\{G_{y}(i, j-1) > G_{y}(i, j+1) \text{ and } G_{y}(i, j) < 0\}$
150	$G_x(i+1, j), G_x(i+1, j+1)$	I: $G_x(i+1, j) < G_x(i+1, j+1)$
degrees	$G_{y}(i,j)$	II: $G_x(i+1, j) > G_x(i+1, j+1)$
165	$G_x(i+1, j), G_x(i+1, j+1), G_x(i+1, j+2)$	I: $G_x(i+1, j) < G_x(i+1, j+2)$
degrees	$G_{y}(i,j)$	II: $G_x(i+1, j) > G_x(i+1, j+2)$

 Table 5.3 Extrema patterns and subtypes of wavelet-reduced ideal step edges

In the following section, we explore the relationship of the ideal step image and its wavelettransformed images. We take the 30-degree ideal step edge image shown in Figure 5.1 for an example. The intensity of the ideal edge varies from 0 to 192. Table 5.4 lists the profiles of its wavelet-transformed images using 9/7-M. The values in the 'ca' block are the intensity values of the approximation coefficients (wavelet-reduced image). The values in the 'ch', 'cv', and 'cd' blocks are the wavelet horizontal, vertical and diagonal detail coefficients respectively. The values in the Gx and Gy blocks are the x and y gradients of the approximation coefficients (the 'ca' block). The extrema of the x- and y- gradients are in bold type as well as the extrema of wavelet horizontal, vertical and diagonal detail coefficients. We always compute the extrema of the x-gradients and wavelet horizontal details along the x direction (in this thesis, the x direction is from up to down, and the y direction is from left to right). And we compute the extrema of the y-gradients and wavelet vertical details along the y direction. The extrema of wavelet diagonal details are computed along 45-degree direction for edges with orientations from 15 degrees to 75 degrees, and computed along 135-degree direction for edges with orientations from 105 degrees to 165 degrees. Since Gx(j) < Gx(j+1) in Table 5.4, it is a subtype I, 30-degree edge.

Ca							cd	j-2	j-1	j	j+1	
0	0	0	2	- 7	-21		0	1	-6	-8	66	-96
0	2	7	-21	6	139	]i-1	-6	-8	66	-96	42	7
7	-21	6	139	207	192	] i	66	-96	42	7	-6	-1
6	139	207	192	189	192	i+1	42	7	-6	-1	0	0
207	192	189	192	192	192		-6	-1	0	0	0	0
189	192	192	192	192	192		0	0	0	0	0.	0
Gx			j	j+1			ch		•	j	j+1	
0	0	0	2	7	-23		0 ·	0	-1	9	27	-104
0	2	7	-23	-1	160	i-1	-1	9	27	-104	82	0
7	-23	-1	160	201	53	i	27	-104	82	0	-12	-2
-1	160	201	53	-18	0	i+1	82	0	-12	-2	0	0
201	.53	-18	0	3	0		-12	-2	0	0	0	0
-18	0	3	0	0	0		0	0	0	0	0	0
Gy			j			-	cv	j-2	j-1	j	j+1	
0	0	0	2	5	-28		0	0	-1	-2	15	-14
0	2	5	-28	_ 27	133	i-1	-1	-2	15	-14	-69	-34
5	-28	27	133	68	-15	i	15	-14	-69	-34	9	2
27	133	68	-15	-3	3	i+1	-69	-34	9	2	-1	0
68	-15	-3	3	0	0		9	2	-1	0	0	0
-3	3	0	0	0	0		-1	0	0	0	0	0

Table 5.4 The profiles of a step edge with 30-degree direction

From our experiments, we have following important results:

(1) The locations of the extrema of wavelet horizontal details 'ch' are always one line up compared to the locations of the extrema of the x-gradients Gx; the locations of the extrema of wavelet vertical details 'cv' are always one line left compared to the locations of the extrema of the y-gradients Gy. Combining the locations of the extrema of the wavelet horizontal and vertical details leads to the locations of the extrema of diagonal details 'cd'. This is due to the fact that the scaling and wavelet functions of the 9/7-M and 9/7-F wavelet transforms are symmetric. It means that the low-pass and band-pass filters of the 9/7-M and 9/7-F wavelet transforms have linear phases. Therefore the locations of these extrema are consistent.

(2) Wavelet horizontal details 'ch' are controlled by the x-gradients Gx; Wavelet vertical details 'cv' are controlled by the y-gradients Gy; Wavelet diagonal details 'cd' are controlled by both the x-gradients Gx and the y-gradients Gy. This is why the zero-degree edge image has all zeros on wavelet vertical details 'cv', wavelet diagonal details 'cd', and y-gradients Gy; and why the 90-degree edge image has all zeros on wavelet horizontal details 'ch', wavelet diagonal details 'ch', wavele

(3) If we reduce or increase the intensity of the ideal step edge image by a certain factor, all the coefficients will also reduce or increase by the same factor in 'ca', 'cd', 'ch', and 'cv', Gx and Gy blocks. This is because of the linear property of the wavelet transforms used in this thesis (the 9/7-M and the 9/7-F). Combining with the result (2), we have: the extrema of wavelet horizontal detail coefficients are proportional to the extrema of x-gradients Gx; the extrema of wavelet vertical detail coefficients are proportional to the extrema of y-gradients Gy; the extrema of wavelet diagonal detail coefficients are proportional to the average extrema of x-gradients Gx and y-gradients Gy. For the example in Table 5.4, we have relationships:

$$ch(i-1, j)/G_x(i, j) = -104/160 = -0.65$$
 (5.3)

$$ch(i-1, j+1)/G_x(i, j+1) = 82/201 = 0.4080$$
 (5.4)

$$cv(i, j-1)/G_{v}(i, j) = -69/133 = -0.5188$$
(5.5)

$$cd(i-1,j) / \frac{[G_x(i,j)+G_y(i,j)]}{2} = -96/(160/2+133/2) = -0.6553$$
(5.6)

Similarly, we can get the relationships between the exterma in the Gx block and the extrema's neighbors in the 'ch' block, between the extrema in the Gy block and the extrema's neighbors in

the 'cv' block, and between the extrema in the Gx and Gy blocks and the extrema's neighbors in the 'cd' block.

Based on above results, if we know the wavelet approximation coefficients of an ideal step edge image, we can use its horizontal and vertical gradients (Gx & Gy) to predict its wavelet horizontal, vertical and diagonal detail coefficients easily and perfectly. We call this method as ideal edge model based prediction (IEMBP).



# 5.2 New Image Interpolation Approach

Figure 5.2 Block diagram of new image interpolation approach

According to the analysis in the last section, a new image interpolation approach is shown in Figure 5.2. It is suitable for two situations: (1) to restore a wavelet-reduced image (the approximation coefficients of the high-resolution image) to its original sizes and (2) to magnify an original image.

The procedures for computing the wavelet detail coefficients from the original image are

- Compute the x-gradients Gx and y- gradients Gy of the original image using equations (5.1);
- (2) Find the extrema of the gradients along the x and y direction using the formula (5.2);
- (3) Scan the Gx and Gy from top to bottom and left to right. If there are extrema at current point either in Gx block or in Gy block, determine whether it is one kind of edges listed in Table 5.3 or not;
- (4) If there is an edge at current point, predict the extrema of wavelet horizontal, vertical and diagonal detail coefficients, and their neighbors according to the edge directions and their subtypes using IEMBP method;

59

(5) Move to next point, and repeat (3) and (4).

Need to mention that a threshold is used in the second step. The value of the threshold depends on the original image and the computational time desired. If all the edges in the original image are very sharp, a big threshold value is preferred. A small threshold generally means more edges will be detected, and thus more computational time is needed.

For any given image, we assume that all detected edges are ideal step edges. In the step four of our new approach, we first initialize all detail coefficients to zeros, and then use IEMBP method to predict the wavelet detail coefficients. Take an example shown in Figure 5.3. 'Ca' block is the original image. We want to predict the 'cd', 'ch', and 'cv' blocks. At the (i, j) location,  $G_x(i, j)$ ,  $G_x(i, j+1)$ ,  $G_y(i, j)$  are maxima. Therefore, there is a 30-degree edge at (i, j). Since  $G_x(i, j) < G_x(i, j+1)$ , this edge belongs to the subtype I. For a 30-degree, subtype I edge, the Matlab code in Figure 5.3 is used to predict the horizontal, vertical and diagonal detail coefficients which are corresponding to the values in Figure 5.3 labeled with '?". The floating numbers used to predict the extrema in the code are from the equations (5.3) ~ (5.6).

Ca							cd	j-2	j-1	j	j+1	
X	X	х	X	X	Х		0	0	0	0	?	0
X	X	X	X	Х	X	]i-1	0	0	0	?	0	0
X	X	X	X	X	Х	i	0	0	?	0	0	0
X	X	Х	X	Х	Х	i+1	0	?	0	0	0	0
X	X	х	X	X	X		?	0	0	0	0	0
X	Х	Х	Х	Х	X		0	0	0	0	0	0
Gx			j	j+1	,	_	ch			j	j+1	
X	X	Х	Х	Х	X		0	0	0	?	?	0
X	X	Х	X	Х	X	]i-1	0	0	0	?	?	0
X	X	X	160	201	X	] i	0	0	0	?	?	0
X	X	X	X	X	X	]i+1	0	0	0	0	0	0
X	X	Х	X	X	X	]	0	0	0	0	0	0
X	X	X	X	X	X		0	0	0	0	0	0
Gy	-		j			-	cv	j-2	j-1	j	j+1	
X	X	X	X	X	X		.0	0	0	0	0	0
X	X	X	X	X	X	i-1	0	0	0	0	0	0
X	X	X	133	x	X	i	?	?	?	?	?	0
X	X	X	X	X	X	i+1	0	0	0	0	0	0
X	X	X	X	X	X		0	0	0	0	0	0
X	X	X	X	X	X	]	0	0	0	0	0	0

60

% P:	redict the	horizontal	detail	coefficients
	☆ extrema			
	ch(i-l,j	) = - Gx(i	,j)*	0.6500;
	ch(i-l,j+	1) = Gx(i	,j+1) *	0.4080;
	<pre>% neighbo</pre>	rs		
	ch(i-2,j	) = Gx(i	,j)*	0.0562;
	ch(i-2,j+	1) = Gx(i	,j+1) *	0.1344;
	ch(i ,j+	1) = -Gx(i)	,j+l) *	0.0597;
* P	redict the	vertical d	etail co	pefficients
	* extremu	<u>In</u>		
	cv(i,j-l)	= - Gy(i,j	) * 0.5	188;
	% neighbo	rs		•
	cv(i,j-3)	= Gy(i,j	) * 0.12	127;
	cv(i,j-2)	= - Gy(i,j	) * 0.10	)53;
	cv(i,j )	= - Gy(i,j	) * 0.2	558;
	cv(i,j+l)	= Gy(i,j	) * 0.00	577;
\$ P	redict the	diagonal d	etail c	oefficients
	Gz = (Gx)	i,j) + Gy(i	;j))/2;	
	<pre>% extremu</pre>	IU		
	cd(i-l,j	) = - Gz *	0.6553	;
	≒ neighbo	rs		
	cd(i+2,j-	3) = - Gz *	0.0410	;
	cd(i+1,j-	2) = Gz *	0.0478	;
	cd(i ,j-	1) = Gz *	0.2865	;
	cd(i-2,j+	1) = Gz *	0.4505	;
	cd(i-3.i+	$(2) = -Gz^*$	0.0546	;

Figure 5.3 The code snippet of the prediction for a 30-degree, subtype I edge

In order to reduce artifacts and get better quality, several strategies are used in our new interpolation approach.

(1) Using selective neighbors of the extrema. Only having the extrema of the horizontal, vertical and diagonal detail coefficients is not enough to produce a good quality image. We have to consider adding more coefficients to smooth the transitions. But how to add these coefficients is a tricky question. Figure 5.4 (a) is produced using all eight neighbors of the extrema. And Figure 5.4 (b) is produced using selective neighbors, i.e. the top and bottom neighbors from the extrema of horizontal detail coefficients, the left and right neighbors from the extrema of vertical detail coefficients, and diagonal neighbors from the extrema of diagonal detail coefficients. The experiment results prove that these selective neighbors can produce edges with better quality than all eight neighbors. Therefore, our new approach predicts selective neighbors only that are shown in the double boxes in Table 5.4. In addition, predicting the selective neighbors only can save computational time.



(a) 8 neighbors

(b) selective neighbors

Figure 5.4 The influence of the neighbors of extrema on an image expanded by a factor 8.

(2) Applying addition operation at the cross point of edges. We often encounter the cross point of two or more edges. How to handle this situation is very important in our new approach. Under this situation, our algorithm will find two or more differently oriented edges at one location. For each orientation edge, we get a set of detail coefficients. The results of our experiment (Figure 5.5) show that adding all sets of detail coefficients together will get better quality images than only using one set of detail coefficients.

(3) Wise edge selector. Edges in images have different slopes. Some have sharp transitions, and some have smooth transitions. Our new approach doesn't work well for the smooth-transiting edges (Figure 5.6 (a)). It is better to exclude these edges. The results of our experiments show that we can still get very good performance even if we do not consider these smooth-transiting edges (Figure 5.6 (b)). We name the module, which drops out smooth-transiting edges, as 'wise edge selector'.

(4) Edges Weighting. A wrong-estimated edge can damage the image quality dramatically. Therefore, our new approach gives a weight (0~1) to each edge based on the length of an edge. For example, if there are 30-degree edges at the locations (i+1, j-2), (i, j), and (i-1, j+2), i.e. 30-degree edges appear three or more times consecutively, a higher weight will be given, or else a low weight will be given. The predicted detail coefficients will be the multiplication of the ideal detail coefficients and the weight.

62

(5) **Specified subtype.** Sometimes, it is difficult to identify the subtype of an edge, especially for the original image. In this situation, a subtype (for example, the subtype I) will be specified.







Figure 5.6 Wise edge selector (a) original image; (b) all edges are processed; (c) wise edge selector is used.

# 5.3 Implementation

In this thesis, Matlab 6.5 is used to develop and test all algorithms. In order to evaluate the performance of the new image interpolation approach, several programs are implemented or existing ones are used as shown below:

• Implemented new image interpolation approaches using 9/7-M and 9/7-F wavelet transforms;

- Implemented wavelet transforms 5/3, 5/11-C, 9/7-M, 9/7-F and 2/10 using the lifting scheme;
- Implemented Kimmel approach [22];
- Implemented Li & Orchard's approach with 4×4 local window [23, 24];
- Bilinear and bicubic interpolation from Matlab;
- MSSIM algorithm from Zhou Wang [13];
- IQM algorithm from the MITRE Corporation [11];
- Other code such as MSE, PSNR, Power Spectrum etc.

## 5.4 The Performance of the Proposed Approach

#### 5.4.1 The Gain of the Proposed Approach

In this section, we will examine the gain of our new approach in the wavelet interpolation method compared with the traditional method that reconstructs images without using the detail coefficients.

**Experiment 1**: Take original image, and reduce it using 9/7-M or 9/7-F wavelet transform, then enlarge the reduced image using inverse wavelet transform. Table 5.5 and Table 5.6 list the PSNR and MSSIM performances of restoration without the detail coefficients and with the detail coefficients predicted by our new approach. Here, the 9/7-M inverse wavelet transform is used to recover the images reduced by the 9/7-M wavelet transform. The same applies to the 9/7-F wavelet transform. In general, the image qualities are improved when using our new approach, especially for images with strong edges. For the "cameraman" image, the PSNR increased by 0.2dB. For other images with various edges, the gains are different. We also notice that the gains from the 9/7-M and 9/7-F wavelet transforms are similar.

Further look into the local strong edges from "cameraman" image. Figure 5.7 shows the restored images with and without our new algorithm. We can see that the quality of the edges from our new approach is indeed much improved (2dB)!

	(9/ <b>7-</b> F	, 9/7-F)	(9/7-M, 9/7-M)			
Image	No details	New approach	No details	New approach		
cameraman	26.93	27.13	26.85	27.06		
lena	30.59	30.74	30.56	30.71		
lily	27.46	27.57	27.41	27.52		
lighthouse	25.97	26.01	25.84	25.87		
bike	26.99	27.09	26.93	27.03		
bird	38.08	38.11	38.28	38.27		
peppers	23.09	23.14	23.01	23.07		
mandrill	17.94	17.93	17.82	17.82		
text	14.64	14.64	14.52	14.53		
text1	33.49	33.28	33.61	33.41		
chestXR	33.93	33.93	33.90	33.90		
pelvisXR	32.71	32.61	32.61	32.52		
brainCT	29.47	29.50	29.43	29.45		
spineCT	30.39	30.38	30.29	30.26		
kidneyUS	32.18	32.13	32.15	32.10		
transUS	32.71	32.64	32.74	32.67		
boneMR	23.41	23.40	23.31	23.31		
cesarMR	32.43	32.19	32.39	32.21		
arteryAng	27.41	27.54	27.27	27.43		
lungAng	32.77	32.77	32.78	32.79		

Table 5.5 The performances of new approach for wavelet-reduced images (PSNR: dB)

	(9/7-F, 9/7-F)		(9/7-M, 9/7-M)			
Image	No details	New approach	No details	New approach		
cameraman	0.8801	0.8831	0.8836	0.8867		
lena	0.9179	0.9189	0.9202	0.9211		
lily	0.9067	0.9082	0.9070	0.9084		
lighthouse	0.8260	0.8271	0.8277	0.8286		
bike	0.8876	0.8893	0.8880	0.8895		
bird	0.9614	0.9616	0.9665	0.9666		
peppers	0.7588	0.7605	0.7595	0.7611		
mandrill	0.7019	0.7022	0.6980	0.6983		
text	0.6021	0.6026	0.5966	0.5972		
text1	0.9687	0.9683	0.9743	0.9739		
chestXR	0.9764	0.9763	0.9799	0.9799		
pelvisXR	0.9532	0.9531	0.9574	0.9574		
brainCT	0.9521	0.9527	0.9544	0.9547		
spineCT	0.9531	0.9533	0.9554	0.9555		
kidneyUS	0.9511	0.9510	0.9525	0.9525		
transUS	0.9445	0.9444	0.9522	0.9521		
boneMR	0.7433	0.7433	0.7405	0.7407		
cesarMR	0.9757	0.9746	0.9787	0.9778		
arteryAng	0.9007	0.9020	0.9024	0.9038		
lungAng	0.9390	0.9388	0.9415	0.9414		

 Table 5.6 The performances of new approach for wavelet-reduced images (MSSIM)

I
**Experiment 2**: Enlarge original image using inverse wavelet transform directly. Table 5.7 and Table 5.8 list the performances of both methods: expanding the original image without the detail coefficients and with the detail coefficients predicted by the new approach. The IQs and Contrasts of all expanded images are increased by using the latter method. Figure 5.8 shows expanded strong edges with and without our new algorithm. Like the case of wavelet-reduced images, our new approach also undoubtedly improves the image qualities of enlarged strong edges for original images. Even magnified by the high factor, the edges are still sharp perpendicular to the contours.



(9/7-M, 9/7-M + new approach)





(9/7-F, 9/7-F + new approach)

Method	RMSE	PSNR (db)	MSSIM
(9/7-M, 9/7-M)	6.52	31.85	0.9597
(9/7-M, 9/7-M + new approach)	5.45	33.40	0.9676
(9/7-F, 9/7-F)	6.54	31.82	0.9540
(9/7-F, 9/7-F + new approach)	4.51	34.31	0.9624

**Figure 5.7** The performances of new approach for the strong edges from a wavelet-reduced image. (*The original image is reduced by a wavelet transform, and then expanded by an inverse wavelet transform. The final image is magnified by a factor of two using the replication method.*)



M - 41 - J	×	2	×4		
Metnoa	IQ	Contrast	IQ	Contrast	
9/7-M	0.627760	0.948154	0.309328	0.958385	
9/7-M + new approach	0.631540	0.948939	0.311269	0.959139	
9/7-F	0.647212	0.956592	0.324151	0.970501	
9/7-F + new approach	0.650312	0.957578	0.326862	0.971561	

**Figure 5.8** The performances of new approach for the strong edges from an original image. (*The original image is expanded once and twice by an inverse wavelet transform.*)

	9/1	7-F	9/7-M		
Image	No details	New approach	No details	New approach	
cameraman	0.1234	0.1250	0.1060	0.1073	
lena	0.1458	0.1471	0.1319	0.1328	
lily	0.2266	0.2287	0.2087	0.2103	
lighthouse	0.1855	0.1881	0.1556	0.1575	
bike	0.2506	0.2536	0.2178	0.2197	
bird	0.0407	0.0409	0.0386	0.0387	
peppers	0.1516	0.1528	0.1230	0.1238	
mandrill	0.4702	0.4736	0.3653	0.3680	
text	0.3863	0.3872	0.3294	0.3297	
text1	0.0351	0.0355	0.0320	0.0323	
chestXR	0.0634	0.0638	0.0599	0.0602	
pelvisXR	0.1071	0.1079	0.0995	0.1000	
brainCT	0.2438	0.2466	0.2346	0.2366	
spineCT	0.1927	0.1938	0.1826	0.1831	
kidneyUS	0.2352	0.2361	0.2224	0.2228	
transUS	0.3667	0.3686	0.3344	0.3357	
boneMR	0.1250	0.1258	0.1041	0.1046	
cesarMR	0.1598	0.1608	0.1519	0.1526	
arteryAng	0.5272	0.5308	0.4928	0.4960	
lungAng	0.0970	0.0970	0.0879	0.0880	

Table 5.7 The performances of new approach for original images (IQ)

 Table 5.8 The performances of new approach for original images (Contrast)

	9/"	7-F	9/7-M		
Image	Image No details New approach		No details	New approach	
cameraman	0.5279	0.5281	0.5222	0.5224	
lena	0.5338	0.5339	0.5281	0.5283	
lily	0.5358	0.5361	0.5284	0.5286	
lighthouse	0.4020	0.4024	0.3920	0.3922	
bike	0.5727	0.5732	0.5613	0.5616	
bird	0.3687	0.3687	0.3673	0.3673	
peppers	0.5791	0.5793	0.5722	0.5724	
mandrill	0.6178	0.6182	0.5932	0.5937	
text	0.3152	0.3155	0.3008	0.3009	
text1	0.1734	0.1735	0.1700	0.1702	
chestXR	0.5498	0.5499	0.5483	0.5483	
pelvisXR	0.4605	0.4607	0.4570	0.4571	
brainCT	0.9814	0.9819	0.9772	0.9776	
spineCT	1.0499	1.0500	1.0470	1.0470	
kidneyUS	1.4092	1.4094	1.4002	1.4002	
transUS	1.1582	1.1585	1.1402	1.1402	
boneMR	0.3618	0.3619	0.3548	0.3549	
cesarMR	0.8245	0.8245	0.8213	0.8213	
arteryAng	1.0662	1.0668	1.0516	1.0523	
lungAng	0.5795	0.5795	0.5751	0.5751	

### 5.4.2 Comparison of Interpolation Approaches for Image Restoration

We know that different wavelet transforms will result in differently reduced images, which will further influence the performance of the restoration. The low-pass filters in the 5/3, 9/7-M and 9/7-F wavelet transform are symmetric (Figure 5.9), and they play important roles in image interpolation. In fact, the wavelet-based image reduction is equivalent to filtering and decimating images. Therefore, we here do an experiment using these low-pass filters. The procedures of this experiment are 1) filter the original image and then downsample the filtered image by two to get the reduced image; 2) expand the reduced image once with different interpolation approaches and 3) compute the PSNR and MSSIM measurements. The purpose of this experiment is to find the best combination(s) of image reduction and image expansion method.



#### Figure 5.9 Several low-pass filters

(L(5/3): low-pass filter of the 5/3 wavelet transform; L(9/7-M): low-pass filter of the 9/7-M wavelet transform; L(9/7-F): low-pass filter of the 9/7-M wavelet transform.)

Figure 5.10 and Figure 5.11 show the performance of several interpolation approaches. The 'None' represents no filter used (i.e. only decimation). All data are averages of the performance of different images. As we expect, overall, the 9/7-M and 9/7-F inverse wavelet transforms using

our new approach have the best performance irrespective of which low-pass filter is used, and then are Kimmel and Li & Orchard's approaches, and Bicubic and Bilinear interpolations are the worst ones.



Figure 5.10 Comparison of interpolation approaches using different low-pass filters (PSNR, average)



Figure 5.11 Comparison of interpolation approaches using different low-pass filters (MSSIM, average)

For the 'cameraman' image, Figure 5.12 and Figure 5.13 show the PSNR and MSSIM measurements of different interpolation methods with different downsampling filters. The best ones are the 9/7-M and 9/7-F inverse wavelet transforms, and then are Kimmel and Li & Orchard's approaches; the worst ones are Bicubic and Bilinear.



Figure 5.12 Comparison of interpolation approaches using different low-pass filters (PSNR, 'Cameraman')



Figure 5.13 Comparison of interpolation approaches using different low-pass filters (MSSIM, 'Cameraman')



(a) Original image



(c) (5/3, Li & Orchard)



(e) (5/3, Bicubic)



(b) (5/3, 9/7-M + new approach)



(d) (5/3, Kimmel)

Figure 5.14 Restored 'Cameraman' images using different interpolation methods. (reduction method, enlargement method)

Figure 5.14 shows restored images using different interpolation methods. In Figure 5.14 (a), 'A' area has strong big edges; 'B' area has strong small edges; 'C' area has weak edges; 'D' area has rich texture. Figure 5.15 also shows enlarged images in the 'A' area. It is clearly seen that our new approach performs the best in these areas. Also, the restored images using our new approach always have great contrasts. The only flaw for our new approach is that it is not smooth enough along some edge contours. The main reason is that most of our ideal edge models are originally jagged. We can improve this problem by constructing better edge models. However, our new approach is still better than the Bicubic approach along these edge contours. Li & Orchard's approach has the best edge contours for these strong big edges though there is a suspicion that it oversmooths raw edges. Kimmel's approach has very good image quality along the strong big edges, but is not very good at strong small edges, weak edges and rich texture.

Based on the above results, we can say that the 9/7-M or 5/3 wavelet transform for image reduction, the 9/7-M inverse wavelet transform with our new approach for image expansion is the best approaches among those image interpolation methods if considering overall image quality.



(d)

(e)

(a) Original (b) (c)

Image	(b)	(c)	(d)	(e)
Method	(5/3, Bicubic)	(5/3, 9/7-M + new approach)	(5/3, Li & Orchard)	(5/3, Kimmel)
PSNR	28.34 dB	33.42 dB	32.39 dB	32.50 dB

**Figure 5.15** Performance comparisons of restored strong big edges (*Images are magnified by a factor of 2 using the replication method*)

### 5.4.3 Comparison of Interpolation Approaches for Image Magnification

In this section, we investigate the performance of our new interpolation approach when it is used to enlarge original images. Since the sizes of the original images and the final images are not the same, the blind image quality measurement is the only choice to evaluate the images besides the subjective measurement. Here IQM measurement is used.

Image	bilinear	bicubic	Kimmel's	Li's	9/7-M (N)	9/7-F (N)
cameraman	0.0889	0.1061	0.0970	0.1137	0.1073	0.1250
lena	0.1194	0.1327	0.1260	0.1397	0.1328	0.1471
lily	0.1907	0.2081	0.2007	0.2155	0.2103	0.2287
lighthouse	0.1283	0.1557	0.1371	0.1833	0.1575	0.1881
bike	0.1869	0.2190	0.1993	0.2349	0.2197	0.2536
bird	0.0365	0.0387	0.0380	0.0397	0.0387	0.0409
peppers	0.1012	0.1204	0.1133	0.1338	0.1238	0.1528
mandrill	0.2739	0.3627	0.3015	0.4115	0.3680	0.4736
text	0.2347	0.3288	0.2701	0.3551	0.3297	0.3872
text1	0.0286	0.0322	0.0307	0.0336	0.0323	0.0355
chestXR	0.0580	0.0604	0.0592	0.0610	0.0602	0.0638
pelvisXR	0.0946	0.0994	0.0974	0.1008	0.1000	0.1079
brainCT	0.2237	0.2335	0.2330	0.2407	0.2366	0.2466
spineCT	0.1736	0.1822	0.1793	0.1874	0.1831	0.1938
kidneyUS	0.2097	0.2225	0.2158	0.2315	0.2228	0.2361
transUS	0.3149	0.3428	0.3220	0.3690	0.3357	0.3686
boneMR	0.0849	0.1038	0.0886	0.1147	0.1046	0.1258
cesarMR	0.1461	0.1528	0.1493	0.1566	0.1526	0.1608
arteryAng	0.4680	0.4961	0.4837	0.5010	0.4960	0.5308
lungAng	0.0810	0.0890	0.0830	0.0918	0.0880	0.0970
Average	-0.0222	0.0000	-0.0131	0.0114	0.0006	0.0238

**Table 5.9** The performances of several interpolation approaches for original images (IQ,  $\times$ 2)

Table 5.10 The performances of several approaches for original images (Contrast, ×2)

Image	bilinear	bicubic	Kimmel's	Li's	9/7-M (N)	9/7-F (N)
cameraman	0.5188	0.5252	0.5218	0.5253	0.5224	0.5281
lena	0.5252	0.5308	0.5287	0.5319	0.5283	0.5339
lily	0.5213	0.5292	0.5273	0.5308	0.5286	0.5361
lighthouse	0.3843	0.3944	0.3863	0.4014	0.3922	0.4024
bike	0.5525	0.5648	0.5568	0.5667	0.5616	0.5732
bird	0.3669	0.3684	0.3681	0.3683	0.3673	0.3687
peppers	0.5669	0.5722	0.5741	0.5745	0.5724	0.5793
mandrill	0.5696	0.5938	0.5852	0.6040	0.5937	0.6182
text	0.2653	0.3018	0.2562	0.3131	0.3009	0.3155
text1	0.1663	0.1707	0.1682	0.1709	0.1702	0.1735
chestXR	0.5498	0.5510	0.5503	0.5510	0.5483	0.5499
pelvisXR	0.4545	0.4571	0.4579	0.4588	0.4571	0.4607
brainCT	0.9659	0.9714	0.9776	0.9796	0.9776	0.9819
spineCT	1.0475	1.0516	1.0499	1.0480	1.0470	1.0500
kidneyUS	1.4086	1.4166	1.4113	1.4123	1.4002	1.4094
transUS	1.1544	1.1730	1.1497	1.1746	1.1402	1.1585
boneMR	0.3486	0.3560	0.3479	0.3617	0.3549	0.3619
cesarMR	0.8214	0.8249	0.8249	0.8251	0.8213	0.8245
arteryAng	1.0533	1.0714	1.0572	1.0554	1.0523	1.0668
lungAng	0.5749	0.5788	0.5757	0.5792	0.5751	0.5795
Average	-0.0094	0.0000	-0.0064	0.0015	-0.0046	0.0034

Table 5.9 and 5.10 show the performances of several interpolation approaches when the magnification factor is two. The 9/7-F inverse wavelet transform with our new approach performs the best, then Li & Orchard's, the 9/7-M, bicubic, Kimmel, and bilinear.

In order to get more accurate evaluation, we look into the 'A', 'B', 'C', and 'D' areas shown in Figure 5.14(a). The magnified images are shown in Figure 5.16. The 9/7-M inverse wavelet transform with our new approach performs the best in all areas. Bicubic method performs well in 'B', 'C', and 'D' areas. Kimmel's method performs well in 'A' and 'C' areas. Li & Orchard's method only perform well in 'A' area, and does have the best contours for strong big edges. These results are consistent with the results from the last section.

How about the image quality if a high magnification factor is applied? Figure 5.17 shows the 16x magnified 45-degree step edge images using different interpolation approaches. We can see: 1) the magnified image using Bicubic approach is badly blurred; 2) Li & Orchard's approach has trouble with this special image; 3) Kimmel' approach does well though the magnified image is blurred a little bit; 4) the 9/7-M with our new approach performs the best with nice-looking edges.

Overall, the 9/7-M inverse wavelet transform with our new approach has the best performance compared with Li & Orchard, Kimmel, bicubic and bilinear when whatever magnification factor is used.



Figure 5.16 Several areas from 2x Enlarged 'Cameraman' image







9/7-M + new approach



## 5.5 Computation Time

The computational time is one important measurement for image interpolation besides image quality. Here, we roughly compare the different approaches since the results from Matlab are not accurate. The test image is 'Cameraman'. We apply same image interpolation method on the same image repeatedly for ten times, and then do the average. The final results are shown in Table 5.11.

Bilinear and Bicubic interpolation methods are fast because they use the Matlab built-in function 'imresize'. Li & Orchard's method is the slowest one because it has to compute the inverse of a matrix. Kimmel's approach has the operation of the square root. Our new approach only has additions and multiplications, and lots of conditional operations and iterative loops, and it is easy to be implemented whether software or hardware. In a word, the computation time from Bicubic, Kimmel, and our new approach should be very close.

Interpolation Methods	Time (seconds)
Bilinear	0.5967 (Matlab built-in function)
Bicubic	0.8182 (Matlab built-in function)
Li & Orchard	44.7173
Kimmel	7.2064
9/7-M + new approach	4.4984 (new approach) +
	4.0128 (inverse wavelet transform)

Table 5.11 Comparison of the computation time for different interpolation methods

## 5.6 Summary

Comparing several image interpolation approaches, we get the following results:

- The 9/7-M inverse wavelet transform with our new approach is the best one when it is used for restoring images that are reduced using the 5/3 or 9/7-M wavelet transform.
- The 9/7-M and 9/7-F based image interpolations achieve the best overall performances.
- The interpolated image using the 9/7-F and 9/7-M with our new approach has the best contrast.

# CONCLUSIONS AND FUTURE RESEARCH

## 6.1 Conclusions

This thesis discusses image interpolation in the space domain and the wavelet domain. The traditional methods such as bilinear and bicubic are very simple but yield the interpolated images blurred. Therefore, edge-based interpolation approaches have been emerging recently. Kimmel's and Li & Orchard's methods much improve the quality of edges in the image, especially the later method. But Li & Orchard's method is computationally expensive, and not suitable for images having small edges and rich texture. Kimmel's approach is also not good for images having small edges and rich texture. The performance of wavelet-based interpolation method is always very high. However, the predicting of the detail coefficients is still an open question. In order to get visually nice images while still keeping a low computational effort, a new approach is proposed in this thesis.

The proposed approach uses the lifting scheme that requires a low computational effort and is easy to implement in hardware or software. The prediction process in the new approach is relatively very simple requiring only additions and multiplications. Our new approach does not need the wavelet decomposition that is a must for most wavelet-based image interpolation approaches. Avoiding the wavelet decomposition introduces two obvious benefits: 1) the computational time of our new approach is relatively low and 2) there is no requirement of the height and width of the original image for image expansion, but for the wavelet decomposition, the size of the image must be a power of 2.

From the experimental results, the 9/7-M inverse wavelet transform with our new approach has the best performance for image expansion. For the case of reducing the image first, and then expanding the reduced image to the original size, the 5/3 and 9/7-M wavelet transform combined with the 9/7-M inverse wavelet transform using our new approach all have very good performances compared with the Bicubic, Kimmel's and Li & Orchard's interpolation methods.

Chapter 6 Conclusion and Future Research

We know the 9/7-M wavelet transform is about 4-times faster than the 9/7-F wavelet transform, which is used often in the literature, and the 5/3 wavelet transform is about 2-times faster than the 9/7-M wavelet transform. Therefore, using the 5/3 wavelet transform for image reduction can save a lot of the computational time. Thus, the best solution is obtained when the 5/3 wavelet transform used in image reduction and the 9/7-M inverse wavelet transform used in image expansion with our new approach.

### 6.2 Future Research

To further improve the quality of the image or extend the application areas for our proposed approach, we here offer the following suggestions:

- Find an effective way to lessen the zigzagging along the strong edges. For example, we can construct better step edge models. Based on our experiments, the zero-degree, 45-degree, 90-degree, and 135-degree edge models perform very well for their nice edges. Other edge models are originally jagged, and perform not well. Therefore, these edge models need to be revised.
- Use more powerful edge detection tools such as Canny Edge Detector which will lead to better edge maps.
- Use non-separable two-dimension wavelet transform such as Contourlet. Nonseparable wavelet transform can make images with better edges. Contourlet can further decompose the image into more directional sub-images not only vertical, horizontal and diagonal ones.
- Extend this approach to color images and video.

- Stéphane Mallat, "A Wavelet Tour of Signal Processing (Second Edition)", Academic Press, 1999.
- [2] Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing (Second Edition)", Addison-Wesley Pub Co, 2002.
- [3] C. Sidney Burrus, Ramesh A. Gopinath, and Haitao Guo, "Introduction to Wavelets and Wavelet Transforms: A Primer", Prentice-Hall, Inc. 1998.
- [4] M. Vetterli and J. Kovacevic, "Wavelets and Subband Coding", Prentice-Hall, Englewood Cliffs, NJ, USA, 1995.
- [5] Milan Sonka and J. Michael Fitzpatrick, "Handbook of Medical Imaging, Volume 2: Medical Image Processing and Analysis (SPIE PRESS Monograph Vol. PM80)", June, 2000.
- [6] Michael D. Adams, "Reversible Integer-to-Integer Wavelet Transforms for Image Coding", PHD Thesis, 2002.
- [7] E. Meijering, "A Chronology of Interpolation: From Ancient Astronomy to Modern Signal and Image Processing", Proceedings of the IEEE, vol. 90, no. 3, pp. 319-342, March 2002.
- [8] BrighamRAD: http://brighamrad.harvard.edu/education/online/tcd/bwh-query-modality.html
- [9] Z. Wang, A.C. Bovik, and L. Lu, "Why is image quality assessment so difficult," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 4, (Orlando), pp. 3313–3316, May 2002.
- [10] N.B. Nill and B.H. Bouzas, "Objective Image Quality Measure Derived from Digital Image Power Spectra", Optical Engineering, vol. 31, pp. 813-825, April 1992.
- [11] The MITRE Corporation: http://www.mitre.org/tech/mtf.
- [12]Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli, "Image quality assessment: From error visibility to structural similarity", IEEE Transactions on Image Processing, vol. 13, no. 4, Apr. 2004.
- [13] SSIM Index Code: http://www.cns.nyu.edu/~zwang/files/research/ssim/ssim\_index.m.

- [14] R.G. Keys, "Cubic Convolution Interpolation for Digital Image Processing", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 29, No. 6, pp. 1153-1160, 1981.
- [15] T. Blu, P. Thévenaz, and M. Unser, "How a Simple Shift Can Significantly Improve the Performance of Linear Interpolation", Proceedings of the IEEE International Conference on Image Processing, Rochester NY, USA, pp. III.377-III.380, Sept. 22-25, 2002.
- [16] P. Thévenaz, T. Blu, M. Unser, "Interpolation revisited [medical images application]", IEEE Transactions on Medical Imaging, Vol. 19, No. 7, pp. 739 – 758, July 2000.
- [17] T.M. Lehmann, C. Gonner, K. Spitzer, "Survey: interpolation methods in medical image", IEEE Transactions on Medical Imaging, Vol. 18, No. 11, pp. 1049 – 1075, Nov. 1999.
- [18]E. Maeland, "On the comparison of interpolation methods", IEEE Transactions on Medical Imaging, Vol. 7, No. 3, pp. 213 – 217, Sept. 1988.
- [19] V.R. Algazi, G.E. Ford, and R. Potharlanka, "Directional interpolation of images based on visual properties and rank order filtering," in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing, vol. 4, pp. 3005–3008, 1991.
- [20] K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images", IEEE Transactions on Image Processing, Vol. 4, No. 3, pp.285--295, March 1995.
- [21] J. Allebach and P.W. Wong, "Edge-directed interpolation," in Proc. IEEE Int. Conf. Image Processing, vol. 3, pp. 707–710, 1996.
- [22] R. Kimmel, "Demosaicing: image reconstruction from color CCD samples", IEEE Transactions on Image Processing, Vol. 8, No. 9, pp. 1221-1228, Sept. 1999.
- [23] X. Li, M.T. Orchard, "New edge-directed interpolation", IEEE Transactions on Image Processing, Vol. 10, No. 10, pp. 1521-1527, Oct. 2001.
- [24] X. Li, M.T. Orchard, "New edge directed interpolation", Proceedings of 2000 International Conference on Image Processing, Vol. 2, pp. 311-314, 10-13 Sept. 2000.
- [25]F.Y. Tzeng, "Adaptive New Edge-Directed Interpolation", ECS231 Course Project, www.cs.ucdavis.edu/~bai/ECS231/finaltzeng.pdf, 2003.
- [26] M. Zhao and G. de Haan, "Content adaptive video up-scaling", Proc ASCI 2003, Heijen, The Netherlands, ISBN 90-803086-8-4, pp. 151-156, June 4-6, 2003.

- [27] J.A. Leitao, M. Zhao and G. de Haan, "Content-adaptive video up-scaling for high-definition displays", IVCP 2003 Proc. Vol. 5022, IIS&T/SPIE Electronic Imaging 2003, Santa Clara, CA, January 2003.
- [28] D.D. Muresan, T.W. Parks, "Optimal Recovery Approach to Image Interpolation", Proceedings of IEEE ICIP 2001, Vol. 3, pp. 848 – 851, 7-10 Oct. 2001.
- [29] D.D. Muresan and T.W. Parks, "Adaptive, Optimal-Recovery Image Interpolation", Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01), Vol. 3, pp. 1949 - 1952, 7-11 May 2001.
- [30] Q. Wang, R.K. Ward, "A New Edge-Directed Image Expansion Scheme", Proceeding of the IEEE International Conference on Image Processing, Vol. III, pp.899 -902, October 2001.
- [31] S.D Bayrakeri, R.M. Mersereau, "A new method for directional image interpolation", Processing of IEEE International Conference on Acoustics, Speech, and Signal (ICASSP), Vol. 4, pp. 2383 - 2386, 9-12 May 1995.
- [32] S. Battiato, G. Gallo, F. Stanco, "A New Edge-Adaptive Algorithm for Zooming of Digital Images", Proceedings of IASTED Signal Processing and Communications SPC 2000, pp. 144-149, Marbella, Spain, Sept. 2000.
- [33] M. Vetterli and C. Herly, "Wavelets and Filter Banks: Theory and Design", IEEE Transactions On Signal Processing, Vol. 40, No. 9, Sept. 1992.
- [34] R.C. Calderbank, I. Daubechies, W. Sweldens, and B. Yeo, "Wavelet Transforms that Map Integers to Integers", Applied and Computational Harmonic Analysis, Vol. 5, No. 3, pp. 332-369, 1998.
- [35] I. Daubechies, W. Sweldens, "Factoring Wavelet Transforms into Lifting Steps", Journal of Fourier Analysis and Applications, Vol. 4, No. 3, pp. 247-269, 1998.
- [36] W. Sweldens, "The lifting scheme: A construction of second generation wavelets", SIAM Journal of Mathematical Analysis, 29(2):511–546, March 1998.
- [37] M. Unser, "Approximation power of biorthogonal wavelet expansions", IEEE Transactions on Signal Processing, Vol. 44, No. 3, pp. 519-527, March 1996.
- [38] M. Unser, "Ten good reasons for using spline wavelets", Proc. SPIE vol. 3169, Wavelet Applications in Signal and Image Processing V, San Diego, CA, pp. 422-431, August 6-9, 1997.

- [39] M. Unser and T. Blu, "Mathematical Properties of the JPEG2000 Wavelet Filters", IEEE Transactions on Image Processing, Vol. 12, No. 9, September 2003.
- [40] Y.L. Huang, R.F. Chang, "MLP Interpolation for Digital Image processing Using Wavelet Transform", Proceedings of IEEE ICASSP-99 International Conference on Acoustics, Speech, and Signal Processing, Phoenix, Arizona, USA, pp.3217-3220.
- [41] K. Kinebuchi, D.D. Muresan, T.W. Parks, "Image Interpolation Using Wavelet-Based Hidden Markov Trees", Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01), Vol. 3, pp. 1957-1960, 7-11 May 2001.
- [42] Y. Zhu, S.C. Schwartz, M.T. Orchard, "Wavelet domain image interpolation via statistical estimation", Proceedings of IEEE International Conference on Image Processing, Vol. 3, pp. 840 - 843, 7-10 Oct. 2001.
- [43] D.D. Muresan and T.W. Parks, "Prediction of Image Detail", Proceedings of IEEE International Conference on Image Processing (ICIP '02), Vol. 2, pp. 323–326, 10-13 Sept. 2000.
- [44] X. Xu, L. Ma, S.H. Soon and T. Chan, "Image Interpolation Based on the Wavelet and Fractal", International Journal of Information Technology, Vol. 7, No. 2, Nov. 2001.
- [45] S. Mallat and S. Zhong, "Characterization of Signals from Multiscale Edges", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 7, pp. 710-732, July 1992.
- [46] S.G. Chang, Z. Cvetkovic, and M. Vetterli, "Resolution Enhancement of Images Using Wavelet Transform Extrema Interpolation", IEEE ICASSP, pp. 2379-2382, May 1995.
- [47] W.K. Carey, D.B. Chuang, S.S. Hemami, "Regularity-Preserving Image Interpolation", IEEE Transactions on Image Processing, Vol. 8, No. 9, pp. 1293-1297, 1999.
- [48] D.D. Muresan and T.W. Parks, "New Image Interpolation Techniques", IEEE 2000 Western New York Image Processing Workshop, Oct. 13, 2000.
- [49] F. Nicolier, O. Laligant, and F. Truchetet, "Multiscale Scheme for Image Magnification", Proceedings of SPIE, Vol. 3974, Image and Video Communications and Processing 2000, pp. 528-536, San Jose, CA, 25-28 January, 2000.
- [50]B. Tao and T. Orchard, "Wavelet-Domain Edge Modeling with Applications to Image Interpolation", F. Nicolier, O. Laligant, and F. Truchetet, "Multiscale Scheme for Image

Magnification", Proceedings of SPIE, Vol. 3974, Image and Video Communications and Processing 2000, pp. 537-548, San Jose, CA, 25-28 January, 2000.

- [51] A. Nosratinia, "Postprocessing of JPEG-2000 images to remove compression artefacts", IEEE Signal Processing Letters, Vol. 10, No. 10, pp. 296 - 299, Oct. 2003.
- [52] M.N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation", IEEE Transactions on Image Processing, August 2003.
- [53] M.N. Do and M. Vetterli, "Contourlets. Beyond Wavelets", G. V. Welland ed., Academic Press, 2003.

# **TEST IMAGES**

Image	Sizes	IQ	Contrast	Description
cameraman	256×256	0.2223	0.5252	Man, camera, grass, distant scene
lena	256×256	0.2791	0.5302	Woman face
lily	230×186	0.3973	0.5306	flower
lighthouse	256×384	0.2694	0.4006	House, fence, tower
bike	512×512	0.4501	0.5661	Man & bike
bird	256×256	0.0855	0.3671	Bird, blurred
peppers	512×512	0.1491	0.4657	Peppers
mandrill	512×512	0.2556	0.3303	Mandrill face, rich texture
text	256×256	0.4674	0.3468	CRT text
text1	640×460	0.0632	0.1708	scanned text, blurred
chestXR	342×288	0.1437	0.5491	X-ray, chest
pelvisXR	320×384	0.2154	0.4575	X-ray, pelvis
brainCT	300×384	0.5177	0.9794	CT, brain
spineCT	410×292	0.3879	1.0524	CT, spine
kidneyUS	432×288	0.4448	1.4024	Ultrasound, kidney
transUS	432×288	0.6740	1.1482	Ultrasound
boneMR	216×212	0.5887	0.6509	MRI, rich texture
cesarMR	284×442	0.3192	0.8236	MRI
arteryAng	386×326	0.9851	1.0691	Angiocardiography, artery
lungAng	318×384	0.1863	0.5772	Angiocardiography, lung

Table A.1 The characteristics of test images

All images are gray, 8-bit depth. The medical images are from BrighamRAD of Harvard University: http://brighamrad.harvard.edu/education/online/tcd/bwh-query-modality.html.

### **Appendix A: Test Images**











peppers



lena

## MT-LEVEL

{spinet}/home/u/rjkroeger/vf: {spinet}/home/u/rjkroeger/vfs Makefile compress.c fi Makefile~ display.c fra {spinet}/home/u/rjkroeger/vfs {spinet}/home/u/rjkroeger/vf: Makefile compress.c Makefile~ display.c fra {spinet}/home/u/rjkroeger/vfs
Makefile display.c im; fileio.c Makefile~ ima fractal.h compress.c im {spinet}/home/u/rjkroeger/vfs
{spinet}/home/u/rjkroeger/vfs
{spinet}/home/u/rjkroeger/vfs





mandrill





lily



text1



lighthouse

## Appendix A: Test Images









pelvisXR

cesarMR





arteryAng



lungAng



brainCT

spineCT

## **APPENDIX B**

## **RESTORED IMAGES (CAMERAMAN)**

(Reduction method, Enlargement method)

Left: restored image;

Right: quality map of the MSSIM measurement



(5/3, 5/3)



(5/11-C, 5/11-C)







(9/7-M, 9/7-M)









(2/10, 2/10)





(Bicubic, Bicubic)





(Decimation, Bilinear)





(Decimation, Bicubic)











(Decimation, Li & Orchard)





(Decimation, 9/7-M + new approach)





(Decimation, 9/7-F + new approach)





(5/3, Bilinear)





(5/3, Bicubic)



(5/3, Kimmel)



(5/3, Li & Orchard)







(5/3, 9/7-M + new approach)





(5/3, 9/7-F + new approach)





(9/7-M, Bilinear)





(9/7-M, Bicubic)





(9/7-M, Kimmel)



(9/7-M, Li & Orchard)







(9/7-M, 9/7-M + new approach)





(9/7-M, 9/7-F + new approach)





(9/7-F, Bilinear)





(9/7-F, Bicubic)











(9/7-F, Li & Orchard)





(9/7-F, 9/7-M + new approach)





(9/7-F, 9/7-F + new approach)


## MAGNIFIED IMAGES (CAMERAMAN)

**Magnification Factor = 2** 



5/3



5/11-C



9/7-M



9/7-F



2/10



9/7-M + new approach



9/7-F + new approach



Bicubic



Bilinear



Kimmel



Li & Orchard