


2008

## Phase-shifting Haar Wavelets For Image-based Rendering Applications

Mais Alnasser  
*University of Central Florida*

 Part of the [Computer Sciences Commons](#), and the [Engineering Commons](#)  
Find similar works at: <https://stars.library.ucf.edu/etd>  
University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact [STARS@ucf.edu](mailto:STARS@ucf.edu).

---

### STARS Citation

Alnasser, Mais, "Phase-shifting Haar Wavelets For Image-based Rendering Applications" (2008).  
*Electronic Theses and Dissertations, 2004-2019*. 3507.  
<https://stars.library.ucf.edu/etd/3507>

# PHASE-SHIFTING HAAR WAVELETS FOR IMAGE-BASED RENDERING APPLICATIONS

by

MAIS ALNASSER

B.S. Jordan University of Science and Technology

M.Sc. University of Michigan-Dearborn

M.Sc. University of Central Florida

A dissertation submitted in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy  
in the School of Electrical Engineering and Computer Science  
in the College of Engineering and Computer Science  
at the University of Central Florida  
Orlando, Florida

Summer Term  
2008

Major Professor:  
Hassan Foroosh

© 2008 Mais Jamil-Odeh Alnasser

## ABSTRACT

In this thesis, we establish the underlying research background necessary for tackling the problem of phase-shifting in the wavelet transform domain. Solving this problem is the key to reducing the redundancy and huge storage requirement in Image-Based Rendering (IBR) applications, which utilize wavelets. Image-based methods for rendering of dynamic glossy objects do not truly scale to all possible frequencies and high sampling rates without trading storage, glossiness, or computational time, while varying both lighting and viewpoint. This is due to the fact that current approaches are limited to precomputed radiance transfer (PRT), which is prohibitively expensive in terms of memory requirements when both lighting and viewpoint variation are required together with high sampling rates for high frequency lighting of glossy material. At the root of the above problem is the lack of a closed-form run-time solution to the nontrivial problem of rotating wavelets, which we solve in this thesis.

We specifically target Haar wavelets, which provide the most efficient solution to solving the triple-product integral, which in turn is fundamental to solving the environment lighting problem. The problem is divided into three main steps, each of which provides several key theoretical contributions. First, we derive closed-form expressions for linear phase-shifting in the Haar domain for one-dimensional signals, which can be generalized to N-dimensional signals due to separability.

Second, we derive closed-form expressions for linear phase-shifting for two-dimensional signals that are projected using the non-separable Haar transform. For both cases, we show that the coefficients of the shifted data can be computed solely by using the coefficients of the original data. We also derive closed-form expressions for non-integer shifts, which has not been reported before. As an application example of these results, we apply the new formulae to image shifting, rotation and interpolation, and demonstrate the superiority of the proposed solutions to existing methods. In the third step, we establish a solution for non-linear phase-shifting of two-dimensional non-separable Haar-transformed signals, which is directly applicable to the original problem of image-based rendering. Our solution is the first attempt to provide an analytic solution to the difficult problem of rotating wavelets in the transform domain.

*Dedicated to my parents Jameel and Rasmieh Alnasser, my brothers Alaa and Fadi and my sister Mariana. Thank you for standing by me.*

## ACKNOWLEDGMENTS

I would like to thank my advisor *Prof. Hassan Foroosh* for his valuable guidance, encouragement and friendship. Starting from my first steps into the Ph.D. program, his generous and sincere support helped me evolve from a student to a scientist.

I would like to thank *Prof. Charles Hughes*, *Prof. Wasfi Mikhael*, *Prof. Sumanta Pattanaik*, and *Prof. Marianna Pensky* for serving as my committee members and for their valuable comments and suggestions.

Special thanks also go to Yuping Shen for his valuable help. I would also like to thank the members of the graphics group for their help and friendship, namely, Mark Colbert, Kevin Boulanger, Musawir A. Shah, Jaakko Kontinen and Juraj Obert. In addition, I would like to thank the previous and current members of the CIL lab, namely, Murat Balci, Alex Cook, Imran Junejo, Brendan Moore, Adeel Buttha and Remo Pillat for their help and encouragement. Finally, I would like to thank Wojciech Matusik for providing the measured BRDF data for the purposes of this research.

# TABLE OF CONTENTS

1	INTRODUCTION . . . . .	1
1.1	THIS CHAPTER: A SUMMARY . . . . .	3
1.2	INTRODUCTION TO WAVELETS . . . . .	4
1.2.1	A LITTLE HISTORY . . . . .	4
1.2.2	WAVELETS FROM A FUNCTIONAL ANALYSIS POINT OF VIEW . . . . .	5
1.2.3	THE THEORY OF MULTIREOLUTION ANALYSIS . . . . .	7
1.2.4	FILTER BANKS AND THE WAVELET TRANSFORM . . . . .	11
1.2.5	A LINEAR-ALGEBRAIC VIEWPOINT . . . . .	15
1.3	IMPORTANT PROPERTIES AND DEFINITIONS . . . . .	17
1.3.1	ORTHOGONALITY AND ORTHONORMALITY . . . . .	17
1.3.2	PERFECT RECONSTRUCTION . . . . .	18
1.3.3	PARSEVAL'S THEOREM AND LOCALIZATION . . . . .	19
1.3.4	SHIFT-INVARIANCE . . . . .	21
1.4	FRAMES AND REDUNDANT WAVELETS . . . . .	22
1.5	COMPLEX WAVELETS . . . . .	24
1.6	PHASE-SHIFTING VS SHIFT-INVARIANCE . . . . .	25



1.7	THE MOTIVATION BEHIND OUR WORK . . . . .	25
1.8	ORGANIZATION OF THE PROPOSAL . . . . .	28
2	PHASE-SHIFTING IN THE DISCRETE HAAR DOMAIN . . . . .	30
2.1	THE ONE-DIMENSIONAL HAAR TRANSFORM TREE . . . . .	30
2.2	SHIFTING FULLY TRANSFORMED SIGNALS . . . . .	34
2.2.1	SHIFTING BY A MULTIPLE OF $2^k$ . . . . .	34
2.2.2	SHIFTING BY AN ODD AMOUNT . . . . .	35
2.2.3	SHIFTING BY AN EVEN AMOUNT THAT IS NOT DIVISIBLE BY $2^k$ . . . . .	36
2.3	SHIFTING PARTIALLY TRANSFORMED SIGNALS . . . . .	39
2.3.1	SHIFTING BY A MULTIPLE OF $2^k$ . . . . .	39
2.3.2	SHIFTING BY AN ODD AMOUNT . . . . .	40
2.3.3	SHIFTING BY AN EVEN AMOUNT THAT IS NOT DIVISIBLE BY $2^k$ . . . . .	41
2.4	NON-INTEGER SHIFTING . . . . .	43
2.5	N-DIMENSIONAL SHIFT . . . . .	47
2.6	COMPLEXITY ANALYSIS . . . . .	47
2.7	EXPERIMENTAL VALIDATION . . . . .	49
2.7.1	IMAGE ROTATION . . . . .	49
3	PHASE-SHIFTING IN THE NON-SEPARABLE HAAR DOMAIN . . . . .	55
3.1	NOTATIONS AND SETUP . . . . .	55
3.2	HORIZONTAL COEFFICIENTS FOR HORIZONTAL SHIFT . . . . .	59

3.2.1	SHIFTING BY A MULTIPLE OF $2^k$	59
3.2.2	SHIFTING BY AN ODD AMOUNT	60
3.2.3	SHIFTING BY AN EVEN AMOUNT THAT IS NOT DIVISIBLE BY $2^k$	62
3.3	VERTICAL COEFFICIENTS FOR HORIZONTAL SHIFT	64
3.4	DIAGONAL COEFFICIENTS FOR HORIZONTAL SHIFT	67
3.5	SHIFTING PARTIALLY-TRANSFORMED SIGNALS	72
3.5.1	SHIFTING BY A MULTIPLE OF $2^k$	73
3.5.2	SHIFTING BY AN ODD AMOUNT	73
3.5.3	SHIFTING BY AN EVEN AMOUNT THAT IS NOT DIVISIBLE BY $2^k$	74
3.6	SUBPIXEL SHIFTING	77
3.7	EXPERIMENTAL RESULTS AND DISCUSSION	84
4	NON-LINEAR PHASE-SHIFTING	87
4.1	THE ENVIRONMENT LIGHTING PROBLEM	88
4.2	THE RENDERING EQUATION	89
4.3	SOLVING THE LIGHT INTEGRAL	91
4.3.1	MONTE CARLO	91
4.3.2	SOLVING THE INTEGRAL IN THE FREQUENCY DOMAIN	93
4.3.3	SPHERICAL HARMONICS	93
4.3.4	HAAR WAVELETS	96
4.4	OUR CONTRIBUTIONS	99
4.5	OUR METHOD	100

4.5.1	ROTATING HAAR COEFFICIENTS . . . . .	101
4.5.2	MATHEMATICAL DESCRIPTION . . . . .	105
4.5.3	ALGORITHMIC OPTIMIZATION . . . . .	107
4.6	EXPERIMENTAL RESULTS AND DISCUSSION . . . . .	109
4.7	CONCLUSION . . . . .	111
LIST OF REFERENCES . . . . .		117

## LIST OF FIGURES

1.1	The Haar wavelets at different scale levels . . . . .	6
1.2	The Box scaling functions at different scale levels . . . . .	8
1.3	Scaling function and wavelet spaces . . . . .	9
1.4	The analysis building block . . . . .	12
1.5	A 4-level wavelet transformation . . . . .	13
1.6	The synthesis building block . . . . .	13
1.7	(a) A signal in the time domain. (b) The signal in time-frequency space. (c) The Fourier-transformed signal in the time-frequency space. (d) The STFT-transformed signal in the time-frequency space. (e) The wavelet-transformed signal in the time-frequency space. . . . .	29

2.1	The Haar coefficients tree contains at each level the scaling and the wavelet coefficients $A_i^l$ and $d_i^l$ , respectively. The Haar transform of the one-dimensional signal $x(n)$ is composed of the dc value $A_0^0$ and the wavelet coefficients $d_i^l$ , where $l = 0, \dots, N - 1$ and $i = 0, \dots, 2^l - 1$ . The scaling coefficients $A_i^l$ at each level are used to derive the analytic expressions for phase-shifting the signal $x(n)$ , but the final expressions are independent of the scaling coefficients. Note that the leaves of the tree, which are composed of the signal $x(n)$ can be considered as the scaling coefficients at level $N$ , namely $A_i^N$ . . . . .	31
2.2	Original image, and the rotated one by 45 degrees using (2.25) with $h = 3$ . . . . .	51
2.3	A magnified portion of the image rotated using equation (2.25). a. integer shift. b. non-integer shift with precision of $\frac{1}{2}$ . c. non-integer shift with precision of $\frac{1}{2^2}$ . d. non-integer shift with precision of $\frac{1}{2^3}$ . e. non-integer shift with precision of $\frac{1}{2^4}$ . f. non-integer shift with precision of $\frac{1}{2^5}$ . . . . .	52
2.4	The above images show the results of successively rotating the original image 16 times by a degree of $\frac{\pi}{8}$ for different methods including ours . . . . .	53
2.5	The above images show the results of successively rotating the original image 16 times by a degree of $\frac{\pi}{8}$ for different methods including ours . . . . .	54
3.1	Two dimensional Haar transform . . . . .	56

3.2	The Haar transform of the two-dimensional signal $x(m, n)$ is composed of the dc value $A_{0,0}^0$ and the wavelet coefficients $a_{i,j}^l$ , $b_{i,j}^l$ and $c_{i,j}^l$ , where $l = 0, \dots, 2^{N-1}$ , $i = 0, \dots, 2^l - 1$ and $j = 0, \dots, 2^l - 1$ . The scaling coefficients $A_{i,j}^l$ at each level are used to help derive the equation for phase-shifting the signal $x(m, n)$ , but are not used in the final form of the equation. Only the first three levels of the tree are shown with one l to avoid cluttering the figure. . . . .	57
3.3	The above figures show the scaling coefficients at level $l = 3$ . In (a), (b), and (c), respectively, the rectangular windows show the coefficients included in evaluating $a_{0,0_{new}}^2$ , $b_{0,0_{new}}^2$ , and $c_{0,0_{new}}^2$ under a horizontal odd shift of one. These are evaluated by summing and subtracting the highlighted quadrants in the windows as shown in the diagrams. . . . .	60
3.4	The above images were phase-shifted in the transform domain and reconstructed to demonstrate the shift. . . . .	72
4.1	A scene rendered using our algorithm. . . . .	87
4.2	Relationship between the incident light and the light leaving a point on an object surface. . . . .	90
4.3	This figure shows an example of evaluating $\frac{dg}{d\phi}$ in equation (4.16) after a rotation along the elevation angle by $20^\circ$ . . . . .	104
4.4	The same scene rendered using our algorithm with and without texture for better showing of shadow. The scene is rendered with aluminum bronze and blue metallic paint measured BRDF's using 64 coefficients. . . . .	113

- 4.5 The above graphs show the rms error comparing rendered images of different models using measured a aluminum bronze material under different environments. The sets of images are generated using preprocessed data under spatial rotation versus our method of rotation. Using our method, we rotated the coefficients at level 6 then evaluated the coefficients at the coarser levels recursively. . . . . 114
- 4.6 The above graphs show the rms error comparing rendered images of different models using measured a blue metallic paint material under different environments. The sets of images are generated using preprocessed data under spatial rotation versus our method of rotation. Using our method, we rotated the coefficients at level 6 then evaluated the coefficients at the coarser levels recursively. . . . . 115
- 4.7 Different degrees of shininess. The rows of images are rendered with 4, 16, 64 and 256 coefficients respectively. Each column is of images is rendered using steel, blue metallic paint and aluminum bronze materials respectively. . . . . 116

# 1 INTRODUCTION

Wavelets have been gaining an ever increasing popularity among scientists and researchers from various disciplines and fields. Many fields have witnessed a migration from the conventional Fourier Transform to the younger Wavelet Transform. Those fields include image processing, signal processing, database systems, computer vision and computer graphics, to mention a few. In signal and image processing, wavelets have been particularly instrumental in methods of constructing “optimal” basis that are often used in various image processing applications, such as classification and identification [CK93, SC94, SC95, DV, DAV99, DV00], and restoration or denoising [CW94, CD95, Don95, FBN07, HW95]. In graphics applications, wavelets have been applied to many problems including level-of-detail (LOD) control for editing and rendering curves and surfaces [FS94, BC90, CG91, FB88, PS83, WW92], surface reconstruction from contours [MSS92], 3D modeling [DJL92][EDD95][HDD94], radiosity and global illumination [CSS96, CW93, GSC93, HSA91, SGC93, SGC94, SS95a] and animation [Coh92]. Wavelets’ ubiquity in so many disciplines is attributed to their fairly simple mathematical ground and the great variety of possible applications in which they can be used.



One of the major applications in computer graphics that started using wavelets, or more specifically Haar wavelets, in the recent few years is Image Based Rendering (IBR). Haar wavelets are being used to represent and compress the light, material and visibility functions required to solve the IBR problem. This is due to their high localization property and their efficiency in dealing with multi-resolution data as opposed to Spherical Harmonics - the Fourier transform analog on the sphere. Furthermore, Haar wavelets provide an efficient solution to the triple-product integral, which is fundamental to solving the lighting integral in IBR applications. However, representing the functions involved in the lighting integral using Haar wavelets, forces the application to store multiple instances of those functions according to their possible and varying orientations. This amounts to gigabytes of redundant data in storage. This is due to the fact that there is no known method for phase-shifting data once the Haar transform is applied. This is referred to in the literature as lack of shift-invariance, which is considered the main disadvantage of using wavelets as opposed to the shift-invariant Fourier Transform and Spherical Harmonics. The ability to phase-shift is commonly intertwined with the shift-invariance property. In this proposal, we present a method for phase-shifting signals without necessarily having the shift-invariance property. For the purpose of IBR applications, the main goal is to be able to shift the data in the Haar transform domain, which does not necessarily require shift-invariance. Basically, achieving phase-shifting, without having shift-invariance, is what results in reducing the storage and producing a more efficient solution to the problem of image-based rendering, which is the main goal of our research.

In this thesis, we aim to establish the grounds for our work by first giving a brief but necessary background to wavelets and their current research state in regards to shift-invariance and phase-

shifting. We then introduce our motivation for the work that we have done to solve the phase-shifting problem for the classical shift-variant Haar wavelets. As mentioned earlier, the lack of a method for phase-shifting wavelets is considered the main disadvantage of using wavelets as opposed to the Fourier Transform and Spherical Harmonics. We present a solution to linear phase-shifting of one-dimensional discrete Haar wavelets, which can be easily generalized to the N-dimensional case. We also present a solution for linear phase-shifting of the two-dimensional non-separable case. These two solutions achieve phase-shifting without having to establish shift-invariance. They also rely solely on the original data and do not introduce any modifications to the classical Haar wavelets. We finally discuss image-based environment lighting, which is a term we use interchangeably with IBR and the necessity for phase-shifting in such an application. We present an innovative method for non-linear phase-shifting of two-dimensional compressed data using the non-separable Haar transform. The phase-shift essentially is a rotation of the spherical data required in the application. Our method provides the first analytic solution to the rotation of Haar wavelets required by image-based environment lighting applications.

## 1.1 THIS CHAPTER: A SUMMARY

The purpose of this chapter is twofold: to provide an introduction to classical wavelets and to motivate our work. The chapter first gives a short background and a history of the evolution of wavelets. It then provides an introduction to wavelets from different viewpoints, namely, functional analysis,

multiresolution theory, and signal processing. We alternate between the different viewpoints as we deem necessary to better explain the design and properties of the wavelet transforms. We then discuss different desired properties including the shift-invariance property and explain their advantages. We discuss two main trends in designing wavelets that aspire to achieve shift-invariance without compromising too many other important properties. Finally, we present the motivation behind our work and detail the organization of this thesis.

More details on the introductory topics covered in this chapter can be found in Burrus, Gopinath and Guo [BGG97], Chui [Chu92], Daubechies [Dau92], Mallat [Mal98], Stollnitz [ES95], Strang and Nguyen [SN97], Vetterli and Kovacevic [VK95] and Vetterli [Vet86].

## 1.2 INTRODUCTION TO WAVELETS

### 1.2.1 A LITTLE HISTORY

The roots of wavelets go back to Karl Weierstrass [Wei95] who in 1873 described a family of fractal functions that are everywhere continuous and nowhere differentiable. These functions are constructed by superimposing scaled copies of a given basis function. In 1910, Alfred Haar constructed an orthonormal system of basis functions now known as the Haar basis [Haa10]. In 1946, Dennis Gabor [Gab46] introduced a non-orthogonal system with unbounded support based on

translated sinusoids modulated by the Gaussian function. However, it wasn't until 1980 that the name “wavelets” emerged in the literature. Goupillaud, Morlet and Grossmann [GGM84, GM84] needed an alternative to Fourier that is more localized in terms of time - We will explain localization in the next section. They replaced the modulation by complex exponentials in the Fourier transform by scaling operations of a prototype function. Localization was achieved by shifting the prototype function over the signal at different scales. They called the prototype function and its different scales “ondelettes”, which means “small wave” in French. Later, it was translated into English as “wavelets”.

### 1.2.2 WAVELETS FROM A FUNCTIONAL ANALYSIS POINT OF VIEW

Wavelets are expansion functions  $\psi_k^j(t)$  that can be used to produce all functions  $f(t) \in L^2(\mathbf{R})$ , where  $L^2(\mathbf{R})$  is the space of square integrable functions:

$$f(t) = \sum_{j,k} b_k^j \psi_k^j(t) \quad (1.1)$$

where  $b_k^j$  are weights called coefficients that decide the contribution of each expansion function towards the reconstruction of  $f(t)$ . Wavelets have the additional characteristic that they are all functions  $\psi_k^j(t)$  that are constructed from a single “prototype” function called the mother wavelet through shifting and dilation. A wavelet  $\psi_k^j(t)$  is a dilated and shifted version of the mother wavelet  $\psi(t)$ , where  $j$  is the number of dilations and  $k$  is the number of shifts. The following formula

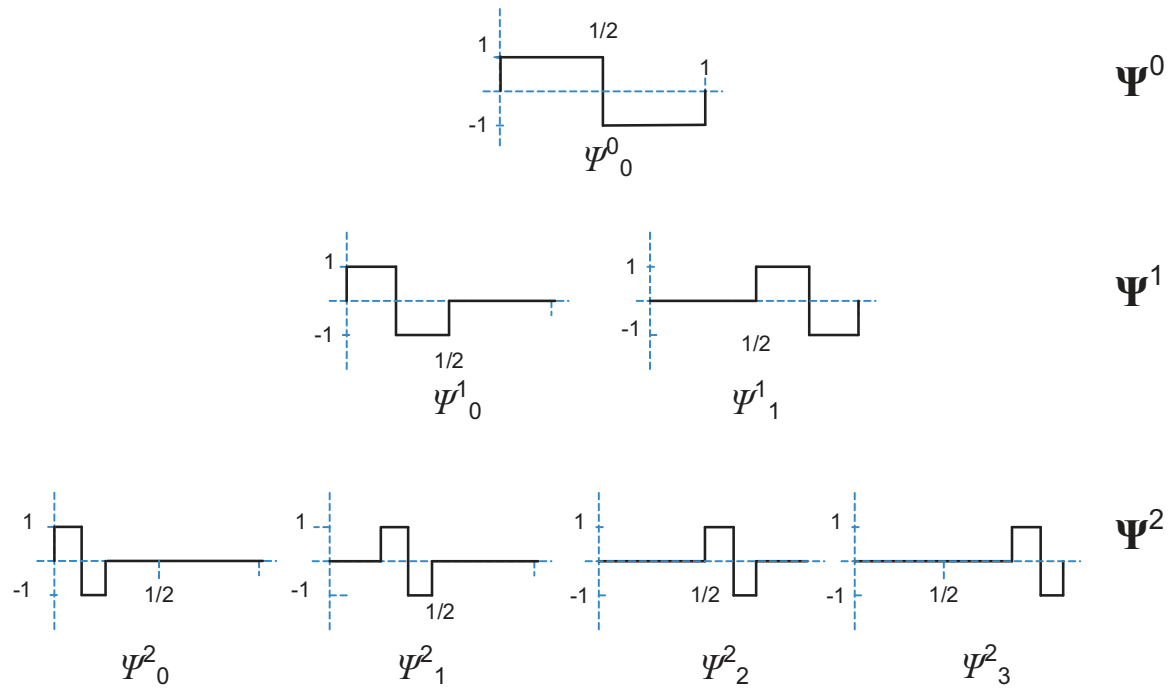


Figure 1.1: The Haar wavelets at different scale levels

explains the relationship between  $\psi_k^j(t)$  and the mother wavelet  $\psi(t)$ :

$$\psi_k^j(t) = \sqrt{2^j} \psi(2^j t - k), \text{ where } \sqrt{2^j} \text{ maintains a constant norm independent of scale } j \quad (1.2)$$

Figure 1.1 shows an example of a mother wavelet, in this case, the Haar mother wavelet, and its daughter wavelets at different scales.

### 1.2.3 THE THEORY OF MULTIREOLUTION ANALYSIS

The different scales of dilation give rise to the theory of multiresolution analysis (MRA) [Mal89]. That is, a signal can be represented at different resolutions allowing for operations like compression and successive reconstruction. Multiresolution analysis is based on the concept of nested spaces  $V^0 \subset V^1 \subset V^2 \subset \dots \subset L^2(\mathbf{R})$ .

As mentioned earlier,  $L^2(\mathbf{R})$  is the space that contains all functions which have a finite, well-defined integral of the square, that is,

$$f(t) \in L^2 \Leftrightarrow \int |f(t)|^2 dt = E < \infty \quad (1.3)$$

$V^j$  has a dimension  $v(j)$  and contains functions of the same size as its dimension.  $V^j$  is defined as the span of a set of linearly independent basis functions called the *Scaling Functions*  $\phi_k^j$ . That is,

$$V^j = \text{span}\{\phi_0^j(t), \phi_1^j(t), \dots, \phi_{v(j)-1}^j(t)\} \quad (1.4)$$

Similar to wavelets, the scaling functions  $\phi_k^j(t)$  are also defined in terms of a parent function  $\phi(t)$  using the following formula:

$$\phi_k^j(t) = \sqrt{2}^j \phi(2^j t - k) \quad (1.5)$$

Figure 1.2 shows an example of a parent scaling function, in this case, the Box function, and its children scaling functions at different scales.

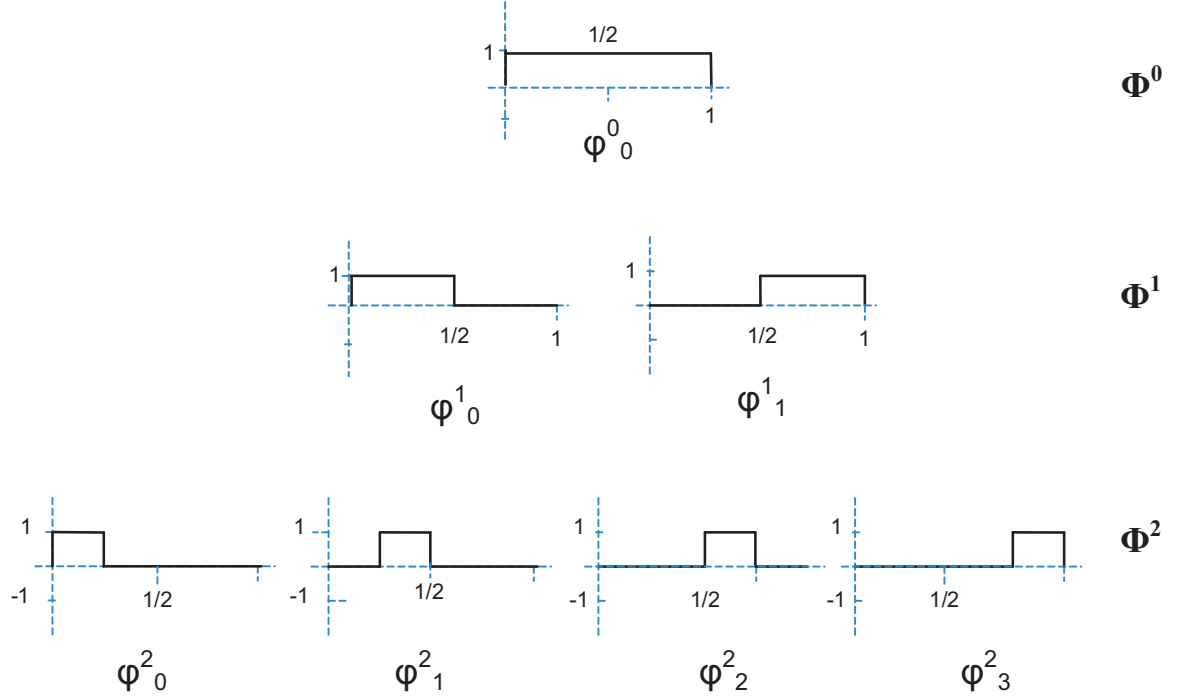


Figure 1.2: The Box scaling functions at different scale levels

The wavelet space  $W^j$  is defined as the complement of  $V^j$  in  $V^{j+1}$  (1.3). That is:

$$V^{j+1} = V^j \oplus W^j \quad (1.6)$$

$L^2$  can now be represented as follows:

$$L^2 = V^0 \oplus W^0 \oplus W^1 \oplus \dots \quad (1.7)$$

or equivalently,

$$L^2 = V^j \oplus W^j \oplus W^{j+1} \oplus \dots \quad (1.8)$$

$j$  can be positive, negative or even  $\infty$  or  $-\infty$ , which allows us to express  $L^2$  as follows:

$$L^2 = V^\infty \quad (1.9)$$

$$= V^{-\infty} \oplus \dots \oplus W^{-2} \oplus W^{-1} \oplus W^0 \oplus W^1 \oplus W^2 \oplus \dots \oplus W^\infty \quad (1.10)$$

where,

$$V^{-\infty} = \{0\} \quad (1.11)$$

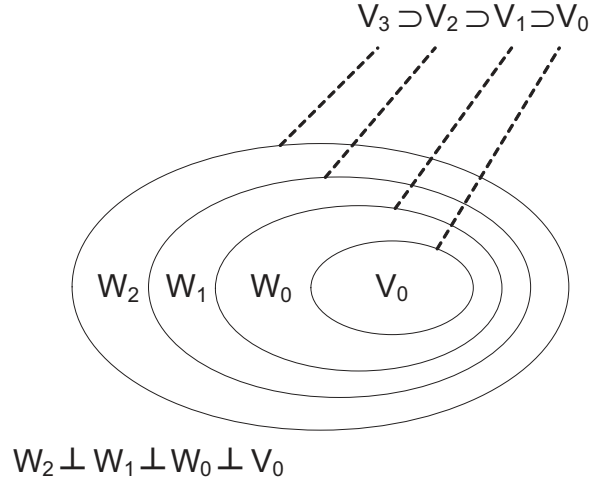


Figure 1.3: Scaling function and wavelet spaces

Note that defining  $W^j$  as the complement of  $V^j$  in  $V^{j+1}$  establishes orthogonality between all functions that belong to  $V^j$  and those that belong to  $W^j$ . In other words, the inner product of  $\phi_k^j$  and  $\psi_l^j$  is equal to zero when  $k \neq l$ :

$$\langle \phi_k^j(t), \psi_l^j(t) \rangle = \int \phi_k^j(t) \psi_l^j(t) dt = 0, \text{ when } k \neq l \quad (1.12)$$



Also, note that the space that contains high resolution signals will contain those of lower resolution also figure (1.3). We can write:

$$f(t) \in V^j \Leftrightarrow f(2t) \in V^{j+1} \quad (1.13)$$

This means that if  $\phi(t)$  is in  $V^0$  then it is also in  $V^1$ , the space spanned by  $\phi(2t)$ . Therefore,  $\phi(t)$  can be expressed in terms of a weighted sum of shifted  $\phi(2t)$  as:

$$\phi(t) = \sum_n h_0(n) \sqrt{2} \phi(2t - n), n \in \mathbf{Z} \quad (1.14)$$

where the coefficients  $h_0(n)$  are a sequence of real or complex numbers called the *Scaling Function Coefficients* and  $\sqrt{2}$  maintains a constant norm.

Similarly, since  $W^0$  is a subset of  $V^1$ ,  $\psi(t)$  can be written as a weighted sum of shifted  $\phi(2t)$  as:

$$\psi(t) = \sum_n h_1(n) \sqrt{2} \phi(2t - n), n \in \mathbf{Z} \quad (1.15)$$

The coefficients  $h_1(n)$  are a sequence of real or complex numbers called the *Wavelet Function Coefficients*. We later discuss how the sequences  $h_0(n)$  and  $h_1(n)$  can be used to construct the wavelet transform.

Following from the above formulation in (1.7), any function  $f(t) \in L^2(\mathbf{R})$  can be written as:

$$f(t) = \sum_{k=-\infty}^{\infty} c_k \phi_k(t) + \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} d_k^j \psi_k^j(t) \quad (1.16)$$

Orthogonality allows coefficients  $c_k$  and  $d_k^j$  to be calculated as the following inner products:

$$c_k = \langle f(t), \phi_k(t) \rangle = \int f(t) \phi_k(t) dt \quad (1.17)$$

$$d_k^j = \langle f(t), \psi_k^j(t) \rangle = \int f(t) \psi_k^j(t) dt \quad (1.18)$$

It also establishes the following simple relationship between  $h_0(n)$  and  $h_1(n)$ :

$$h_1(n) = (-1)^n h_0(1 - n) \quad (1.19)$$

#### 1.2.4 FILTER BANKS AND THE WAVELET TRANSFORM

In the signal processing discipline, the scaling function coefficients  $h_0(n)$  and the wavelet function coefficients  $h_1(n)$  are referred to as lowpass and highpass filters respectively. Convolution of a signal  $x(n)$  with a filter  $h(n)$  is calculated through the following formula:

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \quad (1.20)$$

To perform a one-level transformation on a signal  $x(n) \in V^j$ , the signal is convolved by the lowpass and highpass filters  $h_0(n)$  and  $h_1(n)$  and then downsampled to give the *Scaling Coefficients*  $c^{j-1}$  and *Wavelet Coefficients*  $d^{j-1}$  at level  $j-1$ .

$h_0(n)$  and  $h_1(n)$  are alternatively referred to by their Z-transform equivalents  $H_0(z)$  and  $H_1(z)$ . Filtering in the Z-transform domain becomes a multiplication rather than a convolution, which is a much less expensive process. Figure 1.4 shows the building block for the wavelet transform.

The diagram in figure 1.4 represents the following sequence of operations:

1. Filter an input signal  $x(n) \in V^j$  with the filter whose Z-transform is  $H_0(z)$ .

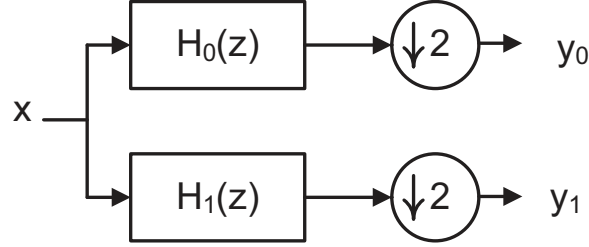


Figure 1.4: The analysis building block

2. Downsample the filter output by 2 to give output coefficients  $y_0(n)$ , or equivalently the scaling coefficients  $c^{j-1}$ .
3. Filter the input signal  $x(n)$  with the filter whose Z-transform is  $H_1(z)$ .
4. Downsample the filter output by 2 to give output coefficients  $y_1(n)$ , or equivalently the wavelet coefficients  $d^{j-1}$ .

To continue transforming the signal, these steps are successively applied to the output of the low-pass filter  $H_0(z)$ , i.e. the scaling coefficients. In other words, a signal  $x(n)$  is transformed by successively applying the highpass and lowpass filters to the scaling coefficients from the previous iteration followed by downsampling at the end of each iteration. Figure 1.5 shows an example of a four-level transformation.

Together, the lowpass and highpass filters form *filter banks*, where the lowpass filter is designed based on a discrete scaling basis  $\phi(n)$  and the highpass filter is designed based on a discrete wavelet basis  $\psi(n)$  [GB92, Vai93]. The lowpass and highpass filters (together with downsampling) sepa-

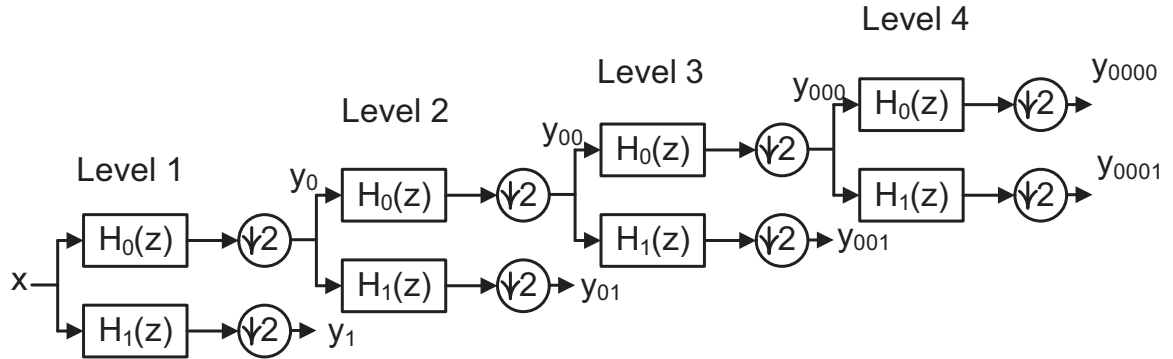


Figure 1.5: A 4-level wavelet transformation

rate the signal into two parts, each half the size of the original signal: a lower resolution part and a complementing part that contains the necessary details for reconstruction.

Another set of lowpass and highpass filters  $g_0(n)$  and  $g_1(n)$  respectively, together with upsampling is used to reconstruct the decomposed signal. The set of filters used for the forward transform is called the *Analysis Filters* and the one used for the inverse transform is called the *Synthesis Filters*. The Z-transform of the lowpass and highpass synthesis filters are denoted by  $G_0(z)$  and  $G_1(z)$ . This notation is used by Vetterli in [Vet86].

The building block for the inverse transform is shown in figure 1.6.

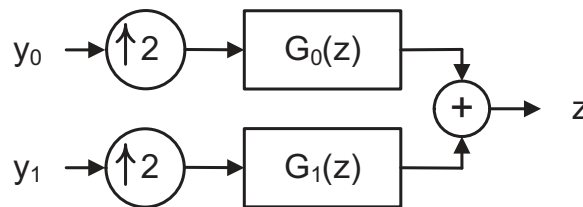


Figure 1.6: The synthesis building block

The building block represents the following operations:

1. Upsample the lowpass coefficients  $y_0$ , or equivalently the scaling coefficients  $c^{j-1}$ , by 2.
2. Filter the upsampled signal with  $G_0(z)$ .
3. Upsample the highpass coefficients  $y_1$ , or equivalently the wavelet coefficients  $d^{j-1}$ , by 2.
4. Filter the upsampled signal with  $G_1(z)$ .
5. Add the two filtered signals together to get the scaling coefficients at level  $V^j$ .

We can use this block repeatedly in order to recover the original sequence from the wavelet transform coefficients:

1. Use the block to reconstruct  $y_{000}$  from  $y_{0000}$  and  $y_{0001}$ .
2. Reconstruct  $y_{00}$  from  $y_{000}$  and  $y_{001}$ .
3. Reconstruct  $y_0$  from  $y_{00}$  and  $y_{01}$ .
4. Reconstruct  $x(n)$  from  $y_0$  and  $y_1$ .

### 1.2.5 A LINEAR-ALGEBRAIC VIEWPOINT

Representing signals as vectors and convolution by filters as matrices allows us to express filtering operations as matrix-vector multiplications [Vet87]. The scaling coefficients can be represented as a vector  $c^j = [c_0^j, \dots, c_{v(j)-1}^j]$ . Similarly, the wavelet coefficients can be represented as a vector  $d^j = [d_0^j, \dots, d_{w(j)-1}^j]$ . Convolution by analysis filters and downsampling can also be expressed by multiplying the coefficients by matrices  $A^j$  and  $B^j$ :

$$c^{j-1} = A^j c^j \quad (1.21)$$

$$d^{j-1} = B^j c^j \quad (1.22)$$

where  $A^j$  is a  $v(j-1) \times v(j)$  dimensional matrix and  $B^j$  is a  $w(j-1) \times v(j)$  dimensional matrix.

On the other hand, convolution by the synthesis filters and upsampling, i.e. the reconstruction operation, can be expressed using matrices  $P^j$  and  $Q^j$  as follows:

$$c^j = P^j c^{j-1} + Q^j d^{j-1} \quad (1.23)$$

$P^j$  and  $Q^j$  are matrices that have  $v(j) \times v(j-1)$  and  $w(j) \times w(j-1)$  dimensions, respectively.

We use the notation:

$$\Phi^j(t) = [\phi_0^j(t), \dots, \phi_{v(j)-1}^j(t)] \quad (1.24)$$

to represent a single row matrix of all scaling basis at level  $j$ . Similarly, we use the notation

$$\Psi^j(t) = [\psi_0^j(t), \dots, \psi_{w(j)-1}^j(t)] \quad (1.25)$$

to represent a single row matrix of all wavelet basis at level  $j$ , where  $v(j)$  and  $w(j)$  are the dimensions of the spaces  $V^j$  and  $W^j$ , respectively. The relationship between spaces  $V^j$  and  $V^{j-1}$  can therefore be expressed through the following equations:

$$\Phi^{j-1}(t) = \Phi^j(t)P^j \quad (1.26)$$

$$\Phi^j(t) = \Phi^{j-1}A^j \quad (1.27)$$

Similarly, the relationship between spaces  $V^j$  and  $W^{j-1}$  can be expressed through the following equations:

$$\Psi^{j-1}(t) = \Phi^j(t)Q^j \quad (1.28)$$

$$\Phi^j(t) = \Psi^{j-1}B^j \quad (1.29)$$

Note that  $\Psi^{j-1}$  can be obtained using the scaling basis  $\Phi^j$ . This is because  $\Psi^{j-1}$ , by its very definition as  $V^{j-1}$ 's complement, is a subspace of  $V^j$  figure (1.3).

Algebraic representation is beneficial for explaining certain properties, such as orthogonality and perfect reconstruction. However, matrix multiplications are avoided whenever implementation is concerned because of their inefficiency.

## 1.3 IMPORTANT PROPERTIES AND DEFINITIONS

As we mentioned earlier, our main goal is to reduce the storage requirement and to improve the efficiency of IBR applications through phase-shifting. The availability of a method for phase-shifting nullifies the need for storage. In this section, we give a more in-depth description of shift-invariance. We also describe and explain certain properties that tend to get sacrificed in the pursuit of phase-shifting by means of achieving shift-invariance. These properties are namely:

- Orthogonality and Orthonormality.
- Perfect Reconstruction.
- Localization.

### 1.3.1 ORTHOGONALITY AND ORTHONORMALITY

A set of wavelet basis functions are orthonormal if the following inner products hold for all  $j, k$  and  $l$ :

$$\langle \phi_k^j | \phi_l^j \rangle = \delta_{k,l} \quad (1.30)$$

$$\langle \psi_k^j | \psi_l^j \rangle = \delta_{k,l} \quad (1.31)$$

$$\langle \phi_k^j | \psi_l^j \rangle = 0 \quad (1.32)$$



If the result of the inner product is the delta function multiplied by a scalar the basis functions are orthogonal. An orthogonal system can be normalized by dividing by the norm  $\|f(t)\| = \langle f|f \rangle$ .

As mentioned earlier, orthogonality allows the scaling coefficients  $c_k$  and the wavelet coefficients  $d_k^j$  to be calculated as the following inner products:

$$c_k = \langle f(t), \phi_k(t) \rangle = \int f(t) \phi_k(t) dt \quad (1.33)$$

$$d_k^j = \langle f(t), \psi_k^j(t) \rangle = \int f(t) \psi_k^j(t) dt \quad (1.34)$$

It also establishes the following simple relationship between filters  $h_0(n)$  and  $h_1(n)$ :

$$h_1(n) = (-1)^n h_0(1-n) \quad (1.35)$$

Furthermore, it implies the following relationships between the analysis and synthesis matrices:

$$A^j = (P^j)^{-1} = (P^j)^T \quad (1.36)$$

$$B^j = (Q^j)^{-1} = (Q^j)^T \quad (1.37)$$

The above relations, in turn, ensure perfect reconstruction.

### 1.3.2 PERFECT RECONSTRUCTION

A system has the perfect reconstruction property if the combination of a forward and reverse wavelet transform leaves any signal unchanged. In other words, perfect reconstruction is the abil-

ity to reconstruct a signal back to its original form without any loss of data. Orthogonality allows for perfect reconstruction but is not mandatory. By examining relations (1.26), (1.27), (1.28) and (1.29), one can easily see that the required relationship between matrices  $P^j$  and  $A^j$ , and between  $Q^j$  and  $B^j$  for a perfect reconstruction is as follows:

$$A^j = (P^j)^{-1} \quad (1.38)$$

$$B^j = (Q^j)^{-1} \quad (1.39)$$

In other words, for a system to have the perfect reconstruction property  $A^j$ ,  $B^j$ ,  $P^j$  and  $Q^j$  have to be invertible.

In signal processing literature, [Vet86] shows that perfect reconstruction is ensured using the following formulae:

$$H_0(z)G_0(z) + H_1(z)G_1(z) = 2 \quad (1.40)$$

$$H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0 \quad (1.41)$$

### 1.3.3 PARSEVAL'S THEOREM AND LOCALIZATION

If the scaling functions and wavelets form an orthonormal basis, the Parseval's theorem can be used to relate the energy of the signal  $f(n)$  to the energy in each of the components and their wavelet

coefficients. Parseval's theorem is as follows:

$$\sum_n |f(n)|^2 = \sum_l |c(l)|^2 + \sum_j \sum_k |d_k^j|^2 \quad (1.42)$$

Parseval's theorem tells us that the signal energy can be partitioned on the time-scale domain, where  $k$  corresponds to time and  $j$  corresponds to scale.

A signal in the time domain is localized with respect to time. That means that at any given time  $t$ , the strength of the signal is known. For example, if the signal is a simple pulse, the location of that pulse is the localization in time. The signal in the time domain is not, however, localized with respect to frequency, that is, one cannot tell what the frequency is at a given time  $t$  because frequency is a function of the entire signal. On the other hand, a signal in the Fourier frequency domain is localized with respect to frequency but not time. In other words, the strength of a given frequency  $f$  is always known but there is no way of telling the times  $t$  at which that frequency occurred. For example, if a Fourier series expansion of a signal has only one large coefficient, then the signal is essentially a single sinusoid at the frequency determined by the index of the coefficient. Diagrams (1.7-b) and (1.7-c) show the localization of energy in the discretized time and Fourier domains.

One of the first attempts to achieve time-localization in the frequency domain was by using the Short-Time (windowed) Fourier Transform (STFT). However, the STFT has a fixed time-frequency resolution of the basis functions (1.7-d), which means that either the time or the frequency resolu-

tion is poor and the trade-off is inflexible. Furthermore, the STFT doesn't lend itself to represent the signal at different scales, therefore, making it unsuitable for multiresolution analysis.

In contrast to Fourier, which is localized only in frequency, and to STFT, which is localized in both frequency and time but is not suitable for multiresolution analysis, the wavelet transform is localized with respect to both frequency and time and is well-suited for MRA. A wavelet representation gives the location in both time and frequency simultaneously. Burrus et al. [BGG97] gives a nice analogy for localization; a wavelet representation is much like a musical score where the location of the notes tells when the tones occur and what their frequencies are figure (1.7-e).

### 1.3.4 SHIFT-INVARIANCE

Shift-invariance is the property where the total energy of the coefficients in any subband is unaffected by translations applied to the original image. Due to localization, which is established through critically sampling the signal and, therefore, violating the Nyquist criterion, shifting the signal causes the energy to redistribute among the different subbands of the transform, unless that shift is a multiple of each of the sampling factors in the system. Simoncelli et al. [SFA92] highlighted the problem of shift-invariance for critically-sampled orthogonal wavelet transforms. They developed a two-dimensional pyramid transform that preserves the energy within each subband by introducing redundancy and relaxing the orthogonality requirement in the system to overcome

the critical sampling. Their decomposition is, however, non-separable and has the disadvantage of non-perfect reconstruction.

Simoncelli et al. [SFA92] were not the first to mention the lack of shift-invariance in the classic critically-sampled wavelet transforms. Strang mentions it as a primary weakness [Str89] and Mallat uses the zero-crossings of the wavelet subbands to create a shift-invariant transform [Mal91].

In the next sections, we discuss the two main trends in the literature to obtain shift-invariance. We also present the progress, advantages and disadvantages of each trend, specifically in terms of the above mentioned properties of orthogonality, perfect reconstruction and localization. Further criteria we look at for evaluating the suitability of those methods for our IBR application are speed, storage cost and preservation of the original design of the wavelet transform. We require no modification to the Haar wavelet transform in specific since no other transform is as efficient in solving the required triple product of the IBR application.

## 1.4 FRAMES AND REDUNDANT WAVELETS

There has been two trends in responding to the shift-invariance requirement. The earlier literature has been focusing on modifying the classical real wavelets to enforce shift-invariance, while attempting to preserve other desired properties. This approach was rediscovered by various authors

independently, and bears different names such as *Algorithme à Trous* [MT89, Dut89, Mal98], Redundant Wavelets [BGG97] and Undecimated Wavelets [LGO96] to name a few.

For a set of functions to be considered as basis it must contain independent elements. In other words, no function in the set can be written as a linear combination of the others. If the set of functions contains functions that are dependent on the others in the same set, the set is then called a frame. A frame is a spanning set, which is not independent. For an expansion set  $\phi_k(t)$  to be a frame for some space, it must satisfy

$$A\|g\|^2 \leq \sum_k |\langle \phi_k, g \rangle|^2 \leq B\|g\|^2, \text{ where } 0 < A, B < \infty \text{ for all signals } g(t) \text{ in the space} \quad (1.43)$$

If  $A = B$  then the frame is a tight frame and

$$A\|g\|^2 = \sum_k \|\langle \phi_k, g \rangle\|^2 \quad (1.44)$$

which is a generalization of the Parseval's theorem.  $A$  is a measure of redundancy for the frame.

Redundant Wavelet Transforms (RWT) have different redundancies at each scale. Except for that fact, RWT's are very much like tight frames, in that they support a form of Parseval's theorem for energy partitioning, which means that they have the time-localization property. The redundancy adds the very desirable shift-invariance property, where the energy is preserved within each subband.

The major drawbacks of this approach, of course, are the undesirable side-effect of overly redundant representation and the high computational cost, since each set of coefficients contains the same number of samples as the input signal. This level of redundancy essentially defeats the purpose of

using wavelets for compressing the functions involved in the lighting integral in IBR applications, and oppose our aim for reducing redundancy.

## 1.5 COMPLEX WAVELETS

In order to alleviate these side-effects, more recently a second approach has been investigated in the literature that attempts to directly construct shift-invariant wavelets. This line of research has led to a new class of wavelets with complex coefficients. Few examples are the Gabor wavelets for texture processing [MM96], harmonic wavelets for vibration and acoustic analysis [New93, New99] and the Complex Wavelet Transform (CWT) for motion estimation [MK96]. In addition to shift-invariance, one particular advantage of complex wavelets is directionality that is similar to the steerable pyramids [SF95]. Complex wavelets prove to be useful in solving the shift-invariance problem without compromising many other properties. However, their major drawbacks are lack of speed and often also poor inversion properties, which means that the signal cannot be reconstructed perfectly. A more successful attempt in this category is perhaps the dual-tree complex wavelet transform (DT-CWT) and its variations [RK99, RK00]. Although, DT-CWT provides a good trade-off between fully decimated wavelets and the redundant wavelet transform, it does so by trading off the compression capabilities and computational time of the classical real wavelets. Furthermore, these wavelets lack the orthogonality property, which is essentially what allows the Haar transform to efficiently provide a solution to the IBR triple product.

## 1.6 PHASE-SHIFTING VS SHIFT-INVARIANCE

The main focus in the wavelet literature has been on establishing phase-shifting by means of establishing shift-invariance. Achieving shift-invariance, however, relies on either introducing redundancy or relaxing orthogonality or the perfect reconstruction properties. These methods all oppose our goal of a more efficient solution to the IBR problem. This leads us to deduce that we have to design our solution for phase-shifting while preserving shift-variance in order not to sacrifice the orthogonality, perfect reconstruction and localization properties of the Haar transform.

In the next section we describe our motivation in more detail.

## 1.7 THE MOTIVATION BEHIND OUR WORK

In the previous sections, we mentioned two main trends in the literature for achieving phase-shifting by means of shift-invariance. The first is introducing redundancy to the transform and the second is using complex transforms. Both trends aim at achieving shift-invariance while not compromising too many other desirable properties that we discussed above. However, redundancy always relaxes the elegant orthogonal property. Complex wavelet transforms trade off the compression capabilities and computational time of the classical real wavelets. Speed, symmetry and separability are other properties that can get sacrificed.



We have initiated and investigated a third line of approach to achieve phase-shifting. Instead of modifying a classical wavelet or introducing a new complex wavelet, our goal is to determine in what way the wavelet coefficients in a fully decimated transform are related to those of a shifted signal. Of course such relation would be wavelet-dependent and may not be a straightforward relation as in redundant wavelets, where the shift in the input results in a shift in the output. The key idea is that as long as the relation is known, one can achieve phase-shifting, since all the coefficients of a shifted signal can be mapped to those of the original signal. On the other hand, we don't require shift-invariance, which means we don't compromise speed, compression or other desired properties. Furthermore, establishing the explicit and direct relations between the coefficients of a signal and its shifted version, would allow us to perform compressed domain processing of signals or images without requiring a chain of forward and backward transforms. This is particularly of interest in applications such as data compression and progressive transmission, or more recent applications in compressed sensing [RN06, MW07, HN05]. Our focus in this dissertation is on the Haar wavelet transform due to additional desirable properties of separability and symmetry. According to [Dau92], there are no wavelets other than Haar that are at once orthogonal, compactly supported and symmetric. Most importantly, our focus on Haar is because of the fact that it provides the most efficient solution to the triple product in IBR applications.

We present a solution to linearly phase-shift the Haar coefficients in the transform domain solely using the available coefficients of the unshifted transformed signal, which we refer to as the 0-shift signal. Our solution generalizes readily to an N-dimensional signal due to separability. We also show how our solution can be extended to non-integer phase shifts. To demonstrate the power of

the proposed approach and to evaluate it, we performed extensive experiments on the problem of accurate image rotation [AF08b, UTY95].

We also present a solution for linearly phase-shifting the 2D non-separable Haar transform. We derive the explicit expressions for shifting the Haar compressed data in the transform domain solely based on using the available coefficients of the original 0-shift signal. We also show how our solution can be expanded for non-integer phase shifts, and evaluate our approach against popular interpolation methods in terms of accumulation of errors through successive shifts [AF08a].

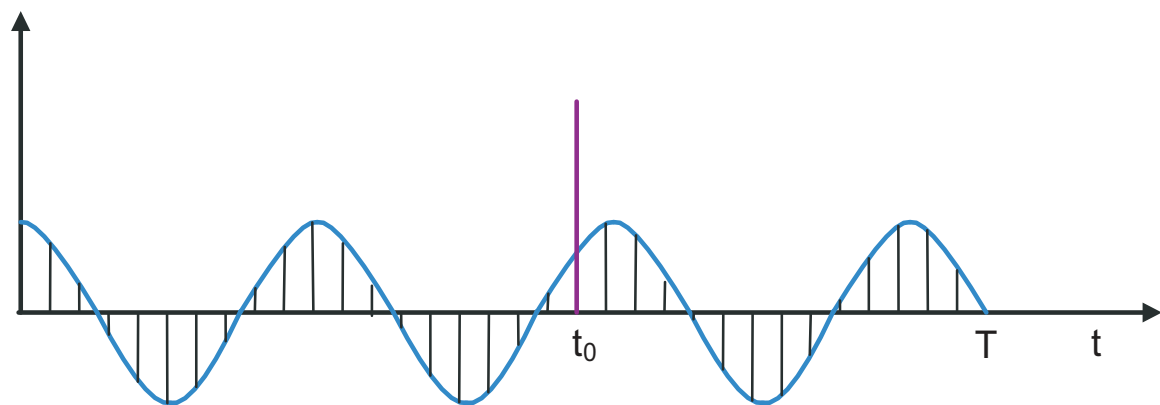
We finally present an innovative method for non-linearly shifting the 2D non-separable Haar wavelets, which is directly applicable to the environment lighting application in Computer Graphics. As we mentioned earlier, Haar wavelets have been used recently to represent and compress the environment lighting, material reflection and the visibility functions involved in the lighting integral due to their proficiency in multi-scale information representation, which translates in the application into different degrees of specularities and shadowing. Haar wavelets proved very useful in solving the lighting integral. A major disadvantage, however, as we also mentioned is the lack of a method for phase-shifting the integral terms in the Haar transform space, which results in forcing the application to store multiple instances of the same data according to its possible and varying orientation. Hence, the usefulness and applicability of our method, which allows us to compute the required information during run-time rather than store it in advance.

## 1.8 ORGANIZATION OF THE PROPOSAL

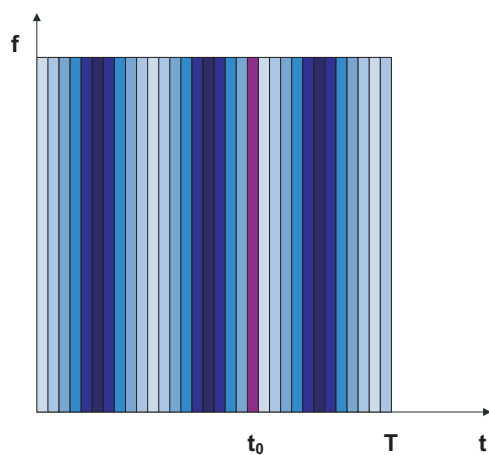
In the next chapter, we first introduce the notations used and briefly describe the Haar transform tree. We then derive our expressions for describing the explicit relations between the Haar coefficients of a 0-shift and a linearly shifted signal for both fully and partially transformed signals. These results are then extended for sub-pixel shifting, followed by full evaluation and testing of the results on image rotation and interpolation problems. The chapter concludes with a brief discussion and some remarks on the proposed new ideas.

In the third chapter, we also start by introducing additional notations and the two-dimensional Haar transform tree. We follow by deriving the formulae for computing the coefficients of a linearly shifted signal. We also show how our method can be also extended to sub-pixel shifting and evaluate the performance of our method against popular interpolation methods in terms of accumulation of errors through successive shifts.

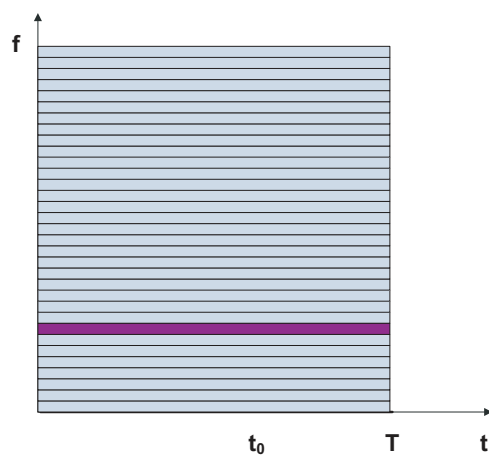
In the last chapter, we introduce the environment lighting problem. We discuss the different methods that have been used so far in synthetic lighting literature. We finally present our solution for non-linear phase-shifting, which we demonstrate by rendering realistic environment-lit synthetic scenes.



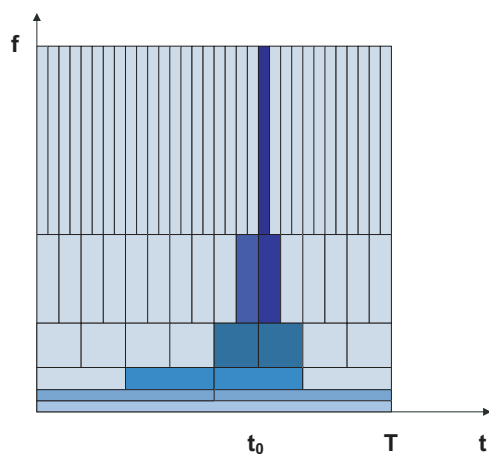
(a)



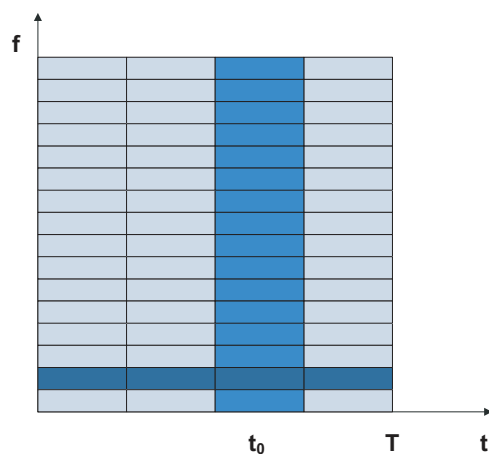
(b)



(c)



(d)



(e)

Figure 1.7: (a) A signal in the time domain. (b) The signal in time-frequency space. (c) The Fourier-transformed signal in the time-frequency space. (d) The STFT-transformed signal in the time-frequency space. (e) The wavelet-transformed signal in the time-frequency space.

## 2 PHASE-SHIFTING IN THE DISCRETE HAAR DOMAIN

### 2.1 THE ONE-DIMENSIONAL HAAR TRANSFORM TREE

Let  $x(n)$  be a one-dimensional signal of size  $2^N$ , where  $N$  is a positive integer. The Haar transform of  $x(n)$ , namely  $H(x(n))$ , has the form:

$$H(x(n)) = \{A_0^0, d_0^0, d_0^1, d_1^1, \dots, d_i^l, \dots, d_0^{N-1}, \dots, d_{2^{N-1}-1}^{N-1}\} \quad (2.1)$$

such that  $A_0^0$  is the dc value of the signal and  $d_i^l$  is the  $i^{\text{th}}$  wavelet coefficient at level  $l$ , where  $l = 0, \dots, N-1$  and  $i = 0, \dots, 2^l - 1$ .

Transforming a signal using Haar wavelets can be easily done by successively convolving the scaled part of the signal by box and Haar filters until the signal is fully transformed (see for instance [VK95] for more details).

We choose to express the Haar transformation using a tree as in Fig. 2.1. The tree is constructed of  $N$  levels with  $x(n)$  residing at the leaves, i.e. the  $N^{\text{th}}$  level. The  $i^{\text{th}}$  node at level  $l$  in the tree

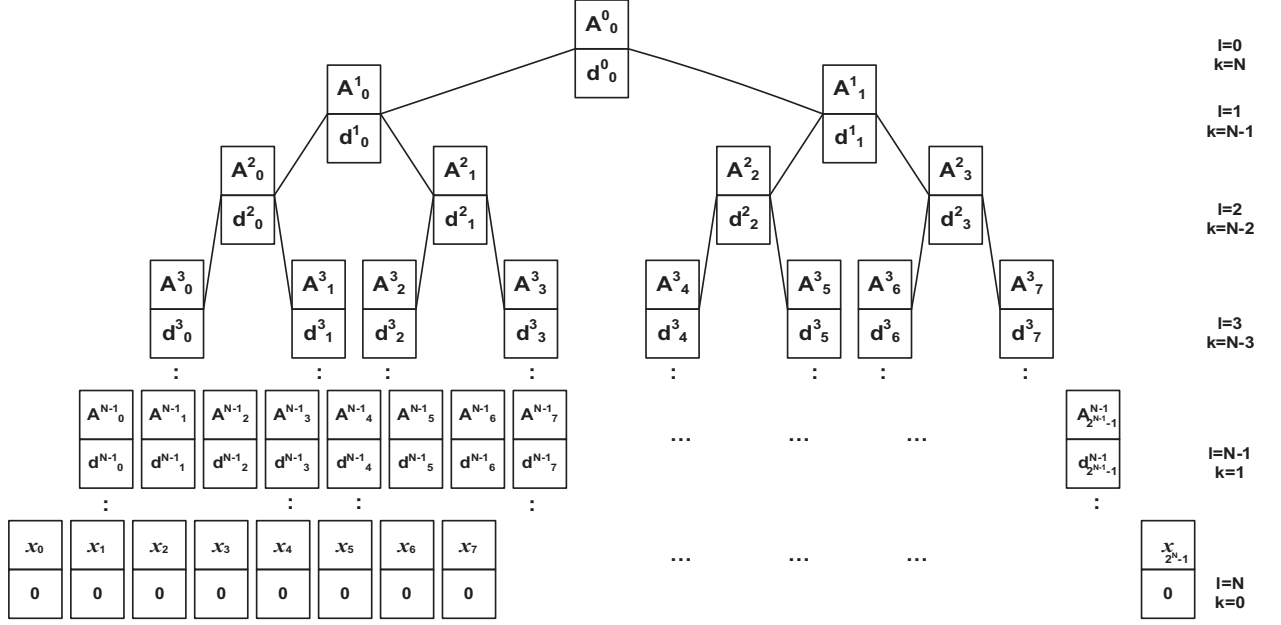


Figure 2.1: The Haar coefficients tree contains at each level the scaling and the wavelet coefficients  $A_i^l$  and  $d_i^l$ , respectively. The Haar transform of the one-dimensional signal  $x(n)$  is composed of the dc value  $A_0^0$  and the wavelet coefficients  $d_i^l$ , where  $l = 0, \dots, N-1$  and  $i = 0, \dots, 2^l - 1$ . The scaling coefficients  $A_i^l$  at each level are used to derive the analytic expressions for phase-shifting the signal  $x(n)$ , but the final expressions are independent of the scaling coefficients. Note that the leaves of the tree, which are composed of the signal  $x(n)$  can be considered as the scaling coefficients at level  $N$ , namely  $A_i^N$ .

can be made to hold the 0-shift  $i^{\text{th}}$  scaling and wavelet coefficients,  $A_i^l$  and  $d_i^l$ , respectively, where  $l = 0, \dots, N-1$  and  $i = 0, \dots, 2^l - 1$ .

Each level in the tree corresponds to a reduction step  $k = 1, \dots, N$ , with the untransformed original signal corresponding to  $k = 0$ . The signal is partially transformed with  $k$  reduction steps if  $0 < k <$

$N$  and is said to be fully transformed if  $k = N$ . At each reduction level  $k$ , one obtains the partially transformed signal  $H^k(x(n))$ .  $H^k(x(n))$  is composed of the scaling coefficients at level  $k$  followed by the wavelet coefficients at the same level and all subsequent reduction levels that are less than  $k$  and greater than 1. That is:

$$H^k(x(n)) = \{A_0^{N-k}, \dots, A_{2^{N-k}-1}^{N-k}, d_0^{N-k}, \dots, d_{2^{N-k}-1}^{N-k}, \dots, d_i^l, \dots, d_0^{N-1}, \dots, d_{2^{N-1}-1}^{N-1}\} \quad (2.2)$$

Where,  $l = N - k, \dots, N - 1$  and  $i = 0, \dots, 2^l$ . Note that  $H^N(x(n)) = H(x(n))$  is the fully transformed signal.

We use the tree to examine the behavior of the wavelet coefficients with respect to shifting. Note that we can denote  $x(i)$  as  $A_i^N$ , in which case  $d_i^N = 0$ . By using this notation,  $l$  now has the range  $0, \dots, N$ . Also, note that the scaling coefficient  $A_i^l$  is related to its parent at level  $l - 1$  by the following relation:

$$A_i^l = \begin{cases} A_{i/2}^{l-1} + d_{i/2}^{l-1}, & i \text{ is even} \\ A_{[i/2]}^{l-1} - d_{[i/2]}^{l-1}, & i \text{ is odd} \end{cases} \quad (2.3)$$

Now, let  $D_i^l$  be the difference between the dc value at the root of the tree  $A_0^0$  and the scaling coefficient  $A_i^l$ . Then

$$A_i^l = A_0^0 + D_i^l \quad (2.4)$$

By substituting (2.4) in (2.3),  $D_i^l$  can be computed recursively solely in terms of the wavelet coefficients using the following relation:

$$D_i^l = \begin{cases} D_{i/2}^{l-1} + d_{i/2}^{l-1}, & i \text{ is even} \\ D_{[i/2]}^{l-1} - d_{[i/2]}^{l-1}, & i \text{ is odd} \\ 0, & l = i = 0 \end{cases} \quad (2.5)$$

It can be verified that  $D_i^l$  can be computed recursively with a complexity of  $O(l)$  for fully-transformed signals, which in itself is very cheap, or be simply tabulated for even a faster retrieval. Also, note that for partially transformed signals, a combination of (2.5) and (2.4) has to be used to evaluate  $D_i^l$ :

$$l = N - k :$$

$$D_i^l = A_i^l - A_0^0$$

$$l > N - k :$$

$$D_i^l = \begin{cases} D_{i/2}^{l-1} + d_{i/2}^{l-1}, & i \text{ is even} \\ D_{[i/2]}^{l-1} - d_{[i/2]}^{l-1}, & i \text{ is odd} \end{cases} \quad (2.6)$$

The complexity for the above equation is even less than that of (2.5) because the recursion needs to go a maximum depth of  $k$  rather than a maximum depth of  $N$ . In other words, the complexity for the above equation is  $O(k - l)$

At level  $N - k$ , there are  $2^k$  non-redundant coefficient sets each of size  $2^{N-k}$  [SB97], where  $k = 1, \dots, N$ . A shift  $s = 0, \dots, 2^N - 1$  can be one of the following possibilities:

- A shift that is divisible by  $2^k$ .



- An odd shift.
- An even shift that is not divisible by  $2^k$ .

In the following sections, we first analyze the behavior of the wavelet coefficients based on the above three possibilities for a fully transformed signal. We then analyze the behavior of the scaling coefficients for signals that are partially transformed. The final analytic solutions that we provide are capable of evaluating the coefficients of the shifted signal solely using the original coefficients of the 0-shift signal, which is the goal of this chapter.

## 2.2 SHIFTING FULLY TRANSFORMED SIGNALS

### 2.2.1 SHIFTING BY A MULTIPLE OF $2^k$

This is the simplest case. A shift  $s$  in the discrete domain that is equal to  $2^k u$  is a circular shift of the 0-shift wavelet coefficients at level  $N - k$  by  $u$ , that is,

$$d_{i_{new}}^{N-k} = d_{(i+u)\%2^{N-k}}^{N-k}, \quad k = 1, \dots, N \quad (2.7)$$

where  $0 \leq u \leq 2^{N-k} - 1$  and  $\%$  is the mod operation. Note that for levels  $N - (k - 1), N - (k - 2), \dots, N - 1$  a shift of  $2^k u$  of the original signal is a circular shift of the coefficients at those levels by  $2u, 2^2 u, \dots, 2^{k-1} u$ , respectively. In other words, a shift of  $2^k u$  of the original signal shifts the

coefficients at level  $N - k$  by  $u$ , while shifting the coefficients at level  $N - (k - 1)$  by twice as much, and the coefficients at level  $N - (k - 2)$  by four times as much and so on.

### 2.2.2 SHIFTING BY AN ODD AMOUNT

By examining the tree in figure (2.1), we notice that:

$$\begin{aligned} d_{i_{new}}^{N-k} &= ((x_{(2^k i + s) \% 2^N} + \dots + x_{(2^{k-1}(2i+1) + s - 1) \% 2^N}) \\ &\quad - (x_{(2^{k-1}(2i+1) + s) \% 2^N} + \dots + x_{(2^k(i+1) + s - 1) \% 2^N}))/2^k \end{aligned} \quad (2.8)$$

In other words,  $d_{i_{new}}^{N-k}$  is the sum of the leaves shifted into its left branch minus the leaves shifted into its right branch divided by  $2^k$ . To simplify the above equation, we set the indices as follows:

$$\begin{aligned} i_1 &= 2^k i + s \\ i_2 &= 2^{k-1}(2i + 1) + s \\ i_3 &= 2^k(i + 1) + s \end{aligned}$$

Using the notation  $A_i^N$  for  $x_i$ , (2.8) now becomes:

$$\begin{aligned} d_{i_{new}}^{N-k} &= ((A_{i_1 \% 2^N}^N + \dots + A_{(i_2 - 1) \% 2^N}^N) \\ &\quad - (A_{i_2 \% 2^N}^N + \dots + A_{(i_3 - 1) \% 2^N}^N))/2^k \end{aligned} \quad (2.9)$$

Substituting (2.4) and then (2.5) in (3.6) and cancelling out the  $A_0^0$ 's, the relation for computing  $d_{i_{new}}^{N-k}$  for a shift  $s$  that is odd becomes:

$$d_{i_{new}}^{N-k} = (D_{i_1 \% 2^{N-1}}^{N-1} + 2 \sum_{m=i_1+1}^{i_2-1} D_{m \% 2^{N-1}}^{N-1} - 2 \sum_{m=i_2+1}^{i_3-1} D_{m \% 2^{N-1}}^{N-1} - D_{i_3 \% 2^{N-1}}^{N-1} - d_{i_1 \% 2^{N-1}}^{N-1} + 2d_{i_2 \% 2^{N-1}}^{N-1} - d_{i_3 \% 2^{N-1}}^{N-1}) / 2^k$$

where,

$$\begin{aligned} i_1 &= 2^{k-1}i + \lfloor s/2 \rfloor \\ i_2 &= 2^{k-2}(2i+1) + \lfloor s/2 \rfloor \\ i_3 &= 2^{k-1}(i+1) + \lfloor s/2 \rfloor \end{aligned} \tag{2.10}$$

Note that for  $k = 1$ ,  $i_2$  would be a non-integer value, in which case we must set  $d_{i_2 \% 2^{N-1}}^{N-1}$  to 0.

### 2.2.3 SHIFTING BY AN EVEN AMOUNT THAT IS NOT DIVISIBLE BY $2^k$

In this case,  $s$  is divisible by  $2^t$ , for  $1 \leq t \leq k-1$  and  $2^t$  is the highest power of 2 by which  $s$  is divisible. This allows us to let  $s = 2^t u$ , where  $0 \leq u \leq 2^{N-t} - 1$ . This means that the coefficients at levels  $N-1, \dots, N-t$  follow the first case. In other words, the 0-shift coefficients at levels  $N-1, N-2, \dots, N-t$  are circularly shifted by  $2^{t-1}u, 2^{t-2}u, \dots, u$ , respectively. Since  $2^t$  is the highest power of 2 by which  $s$  is divisible,  $u$  must be odd. This allows us to treat this case as an odd shift of the scaling coefficients at level  $N-t$ . In other words, at level  $N-k$ ,  $d_{i_{new}}^{N-k}$  can be

evaluated using the following modification of equation (3.6):

$$\begin{aligned} d_{i_{new}}^{N-k} &= ((A_{i_1 \% 2^{N-t}}^{N-t} + \dots + A_{(i_2)-1 \% 2^{N-t}}^{N-t}) \\ &\quad - (A_{i_2 \% 2^{N-t}}^{N-t} + \dots + A_{(i_3-1) \% 2^{N-t}}^{N-t}))/2^{k-t} \end{aligned}$$

where,

$$\begin{aligned} i_1 &= 2^{k-t}i + s/2^t \\ i_2 &= 2^{k-t-1}(2i+1) + s/2^t \\ i_3 &= 2^{k-t}(i+1) + s/2^t \end{aligned} \tag{2.11}$$

Following the same steps, the above can be rewritten as:

$$\begin{aligned} d_{i_{new}}^{N-k} &= (D_{i_1 \% 2^{N-t-1}}^{N-t-1} + 2 \sum_{m=i_1+1}^{i_2-1} D_{m \% 2^{N-t-1}}^{N-t-1} \\ &\quad - 2 \sum_{m=i_2+1}^{i_3-1} D_{m \% 2^{N-t-1}}^{N-t-1} - D_{i_3 \% 2^{N-t-1}}^{N-t-1} \\ &\quad - d_{i_1 \% 2^{N-t-1}}^{N-t-1} + 2d_{i_2 \% 2^{N-t-1}}^{N-t-1} - d_{i_3 \% 2^{N-t-1}}^{N-t-1})/2^{k-t} \end{aligned}$$

where,

$$\begin{aligned} i_1 &= 2^{k-t-1}i + \lfloor s/2^{t+1} \rfloor \\ i_2 &= 2^{k-t-2}(2i+1) + \lfloor s/2^{t+1} \rfloor \\ i_3 &= 2^{k-t-1}(i+1) + \lfloor s/2^{t+1} \rfloor \end{aligned}$$

and,

$$d_{i_2 \% 2^{N-1}}^{N-1} = 0, \text{ if } i_2 \text{ is non-integer.} \tag{2.12}$$

Note that the second case is the same as the third case when  $t = 0$ . That leaves us with the following formula:

$k > t :$

$$\begin{aligned} d_{i_{new}}^{N-k} = & (D_{i_1 \% 2^{N-t-1}}^{N-t-1} + 2 \sum_{m=i_1+1}^{i_2-1} D_{m \% 2^{N-t-1}}^{N-t-1} \\ & - 2 \sum_{m=i_2+1}^{i_3-1} D_{m \% 2^{N-t-1}}^{N-t-1} - D_{i_3 \% 2^{N-t-1}}^{N-t-1} \\ & - d_{i_1 \% 2^{N-t-1}}^{N-t-1} + 2d_{i_2 \% 2^{N-t-1}}^{N-t-1} - d_{i_3 \% 2^{N-t-1}}^{N-t-1}) / 2^{k-t} \end{aligned}$$

$k \leq t :$

$$d_{i_{new}}^{N-k} = d_{(i+s/2^k) \% 2^{N-k}}^{N-k}$$

where,

$$\begin{aligned} i_1 &= 2^{k-t-1}i + \lfloor s/2^{t+1} \rfloor \\ i_2 &= 2^{k-t-2}(2i+1) + \lfloor s/2^{t+1} \rfloor \\ i_3 &= 2^{k-t-1}(i+1) + \lfloor s/2^{t+1} \rfloor \end{aligned}$$

and,

$$d_{i_2 \% 2^{N-1}}^{N-t-1} = 0, \text{ if } i_2 \text{ is a non-integer} \quad (2.13)$$

The above relation can now be used to evaluate the new wavelet coefficients of the Haar transform at all different levels after any shift  $s = 0, \dots, 2^N - 1$  using only the coefficients of the 0-shift signal. The worst case complexity for evaluating  $d_{i_{new}}^{N-k}$  using (2.13) is  $O(\log(L))$ , where  $L$  is the size of the signal  $x(n)$  (see the complexity analysis section for more details).

## 2.3 SHIFTING PARTIALLY TRANSFORMED SIGNALS

Depending on the application, the original signal might not be fully transformed. As we mentioned earlier, a signal that has  $k$  degrees of reduction has the form:

$$H^k(x(n)) = \{A_0^{N-k}, \dots, A_{2^{N-k}-1}^{N-k}, d_0^{N-k}, \dots, d_{2^{N-k}-1}^{N-k}, \dots, d_i^l, \dots, d_0^{N-1}, \dots, d_{2^{N-1}-1}^{N-1}\} \quad (2.14)$$

Where,  $1 \leq k \leq N-1$ ,  $l = N-k, \dots, N-1$  and  $i = 0, \dots, 2^l$ .

A signal that is partially transformed is composed of both scaling coefficients and wavelet coefficients. Equation (2.13) shows how to evaluate the wavelet coefficients of a fully transformed shifted signal, which also applies to evaluating the wavelet coefficients of a partially transformed signal. In this section we show how to evaluate the scaling coefficients at reduction step  $k$  for a signal that has been decomposed  $k$  times and shifted by the integer amount  $s$  in the time domain.

### 2.3.1 SHIFTING BY A MULTIPLE OF $2^k$

Similar to evaluating the wavelet coefficients case, a shift  $s$  in the discrete domain that is equal to  $2^k u$  is a circular shift of the 0-shift scaling coefficients at level  $N-k$  by  $u$ , that is,

$$A_{i_{new}}^{N-k} = A_{(i+u)\%2^{N-k}}^{N-k}, \quad k = 1, \dots, N-1 \quad (2.15)$$

where  $0 \leq u \leq 2^{N-k} - 1$ .

### 2.3.2 SHIFTING BY AN ODD AMOUNT

By examining the tree in figure (2.1), we notice that:

$$\begin{aligned}
 A_{i_{new}}^{N-k} = & ((x_{(2^k i + s) \% 2^N} + \dots + x_{(2^{k-1}(2i+1) + s - 1) \% 2^N}) \\
 & + (x_{(2^{k-1}(2i+1) + s) \% 2^N} + \dots + x_{(2^k(i+1) + s - 1) \% 2^N})) / 2^k
 \end{aligned}
 \tag{2.16}$$

In other words,  $A_{i_{new}}^{N-k}$  is the sum of the leaves shifted into its left branch plus the leaves shifted into its right branch divided by  $2^k$ . To simplify the above equation, we use only the starting and ending coefficients and we also use the notation  $A_i^N$  for  $x_i$ :

$$A_{i_{new}}^{N-k} = (A_{i_1 \% 2^N}^N + \dots + A_{(i_2 - 1) \% 2^N}^N) / 2^k
 \tag{2.17}$$

Where,

$$\begin{aligned}
 i_1 &= 2^k i + s \\
 i_2 &= 2^k (i + 1) + s
 \end{aligned}$$

Substituting (2.3) in the above, we get

$$A_{i_{new}}^{N-k} = (A_0^0 + D_{i_1 \% 2^N}^N + \dots + A_0^0 + D_{(i_2 - 1) \% 2^N}^N) / 2^k
 \tag{2.18}$$

The number of  $A_0^0$ 's is equal to the number of coefficients  $A_i^l$  being summed, which is equal to  $2^k$ .

We factor out  $A_0^0$ :

$$A_{i_{new}}^{N-k} = A_0^0 + (D_{i_1 \% 2^N}^N + \dots + D_{(i_2 - 1) \% 2^N}^N) / 2^k
 \tag{2.19}$$

Substituting (2.5) and simplifying, we get the analytic solution for evaluating  $A_{i_{new}}^{N-k}$  under an odd shift  $s$ :

$$A_{i_{new}}^{N-k} = A_0^0 + (D_{i_1 \% 2^{N-1}}^{N-1} + 2 \sum_{m=i_1+1}^{i_2-1} D_{m \% 2^{N-1}}^{N-1} + D_{i_2 \% 2^{N-1}}^{N-1} - d_{i_1 \% 2^{N-1}}^{N-1} + d_{i_2 \% 2^{N-1}}^{N-1})/2^k$$

where,

$$\begin{aligned} i_1 &= 2^{k-1}i + \lfloor s/2 \rfloor \\ i_2 &= 2^{k-1}(i+1) + \lfloor s/2 \rfloor \end{aligned} \tag{2.20}$$

### 2.3.3 SHIFTING BY AN EVEN AMOUNT THAT IS NOT DIVISIBLE BY $2^k$

For a shift  $s = 2^t u$ , where  $0 \leq u \leq 2^{N-t} - 1$  and  $t < k$ , we can treat this case as an odd shift of the coefficients at level  $N - t$ , which is similar to what we did in evaluating the wavelet coefficients under a shift  $s = 2^t u$ .  $A_{i_{new}}^{N-k}$  can now be evaluated using the following equation:

$$A_{i_{new}}^{N-k} = (A_{i_1 \% 2^{N-t}}^{N-t} + \dots + A_{(i_2-1) \% 2^{N-t}}^{N-t})/2^{k-t} \tag{2.21}$$

Proceeding as we did in the odd shift case, we get the following solution:



$$A_{i_{new}}^{N-k} = A_0^0 + (D_{i_1 \% 2^{N-t-1}}^{N-t-1} + 2 \sum_{m=i_1+1}^{i_2-1} D_{m \% 2^{N-t-1}}^{N-t-1} + D_{i_2 \% 2^{N-t-1}}^{N-t-1} - d_{i_1 \% 2^{N-t-1}}^{N-t-1} + d_{i_2 \% 2^{N-t-1}}^{N-t-1}) / 2^{k-t}$$

where,

$$\begin{aligned} i_1 &= 2^{k-t-1}i + \lfloor s/2^{t+1} \rfloor \\ i_2 &= 2^{k-t-1}(i+1) + \lfloor s/2^{t+1} \rfloor \end{aligned} \quad (2.22)$$

Combining the three cases, the final result becomes:

$k > t :$

$$A_{i_{new}}^{N-k} = A_0^0 + (D_{i_1 \% 2^{N-t-1}}^{N-t-1} + 2 \sum_{m=i_1+1}^{i_2-1} D_{m \% 2^{N-t-1}}^{N-t-1} + D_{i_2 \% 2^{N-t-1}}^{N-t-1} - d_{i_1 \% 2^{N-t-1}}^{N-t-1} + d_{i_2 \% 2^{N-t-1}}^{N-t-1}) / 2^{k-t}$$

$k \leq t :$

$$A_{i_{new}}^{N-k} = A_{(i+s/2^k) \% 2^{N-k}}^{N-k}$$

where,

$$\begin{aligned} i_1 &= 2^{k-t-1}i + \lfloor s/2^{t+1} \rfloor \\ i_2 &= 2^{k-t-1}(i+1) + \lfloor s/2^{t+1} \rfloor \end{aligned} \quad (2.23)$$

The above relation can now be used to evaluate the new scaling coefficients of a partially transformed signal with  $k$  reduction steps after any shift  $s = 0, \dots, 2^N - 1$  using only the coefficients of

the 0-shift signal. The worst case complexity for evaluating  $A_{i_{new}}^{N-k}$  using (2.23) is  $O(\log(L))$ , where  $L$  is the size of the signal  $x(n)$  (see the complexity analysis section for more details).

## 2.4 NON-INTEGER SHIFTING

In this section, we show how our solution can be extended to achieve non-integer shifts. Although, our model is based on up-sampling the original signal, the final relations that are derived require using only the coefficients of the original signal. Up-sampling by a factor of 2 can be modeled as adding levels to the lowest part of the transform tree and setting the wavelet coefficients in those levels to zero, with the lowest level being  $N - 1$ . On the other hand, shifting the up-sampled signal by an amount  $u$  is equivalent to shifting the original signal by  $\frac{u}{2}$ , which is a precision of  $\frac{1}{2}$ . More generally, adding  $h$  levels would enable us to obtain a precision of  $\frac{1}{2^h}$ .

Let the size of the signal be  $2^N$ ,  $N' = N + h$  and  $k = 1 + h, \dots, N + h$ , where  $h$  is the number of added levels. Equation (2.13) can now be modified to allow for non-integer shifting by a precision of  $\frac{1}{2^h}$

as follows:

$k > t$  :

$$\begin{aligned}
d_{i_{new}}^{N'-k} &= (D_{i_1 \% 2^{N'-t-1}}^{N'-t-1} + 2 \sum_{i_1+1}^{i_2-1} D_{m \% 2^{N'-t-1}}^{N'-t-1} \\
&\quad - 2 \sum_{i_2+1}^{i_3-1} D_{m \% 2^{N'-t-1}}^{N'-t-1} - D_{i_3 \% 2^{N'-t-1}}^{N'-t-1} \\
&\quad - d_{i_1 \% 2^{N'-t-1}}^{N'-t-1} + 2d_{i_2 \% 2^{N'-t-1}}^{N'-t-1} - d_{i_3 \% 2^{N'-t-1}}^{N'-t-1}) / 2^{k-t}
\end{aligned}$$

$k \leq t$  :

$$d_{i_{new}}^{N'-k} = d_{(i+s/2^k) \% 2^{N'-k}}^{N'-k}$$

where,

$$\begin{aligned}
i_1 &= 2^{k-t-1}i + \lfloor s/2^{t+1} \rfloor \\
i_2 &= 2^{k-t-2}(2i+1) + \lfloor s/2^{t+1} \rfloor \\
i_3 &= 2^{k-t-1}(i+1) + \lfloor s/2^{t+1} \rfloor
\end{aligned}$$

and,

$$d_{i_2 \% 2^{N'-1}}^{N'-t-1} = 0, \text{ if } i_2 \text{ is a non-integer} \quad (2.24)$$

On the other hand, we can verify that  $D_i^{N+h_0} = D_{\lfloor i/2^{h_0} \rfloor}^N$ , where  $0 \leq h_0 \leq h$ . Using (2.5), we also know that:

$$D_i^N = \begin{cases} D_{i/2}^{N-1} + d_{i/2}^{N-1}, & i \text{ is even} \\ D_{\lfloor i/2 \rfloor}^{N-1} - d_{\lfloor i/2 \rfloor}^{N-1}, & i \text{ is odd} \end{cases}$$

The above result allows us to modify (2.24) in such a way that avoids having to up-sample the signal for non-integer shifts, saving thus memory space in actual implementation, especially that

the size increases exponentially. However, We have to split the equation into two cases. The first is when  $h \geq t + 1$ , which is when the coefficients at the added levels are being used to evaluate  $d_{i_{new}}^{N'-k}$ . The second is when  $t$  is large enough for the coefficients at the original levels of the tree to

be used. This leads to the new form of the phase shifting relation for non-integer values as follows:

$$h \geq t + 1 :$$

$$d_{i_{new}}^{N'-k} = (D_{\lfloor \frac{i_1 \% 2^{N'-t-1}}{2^{N'-t-t}} \rfloor}^N + 2 \sum_{i_1+1}^{i_2-1} D_{\lfloor \frac{m \% 2^{N'-t-1}}{2^{N'-t-t}} \rfloor}^N - 2 \sum_{i_2+1}^{i_3-1} D_{\lfloor \frac{m \% 2^{N'-t-1}}{2^{N'-t-t}} \rfloor}^N - D_{\lfloor \frac{i_3 \% 2^{N'-t-1}}{2^{N'-t-t}} \rfloor}^N) / 2^{k-t}$$

$$h < t + 1 :$$

$$k > t :$$

$$d_{i_{new}}^{N'-k} = (D_{i_1 \% 2^{N'-t-1}}^{N'-t-1} + 2 \sum_{i_1+1}^{i_2-1} D_{m \% 2^{N'-t-1}}^{N'-t-1} - 2 \sum_{i_2+1}^{i_3-1} D_{m \% 2^{N'-t-1}}^{N'-t-1} - D_{i_3 \% 2^{N'-t-1}}^{N'-t-1} - d_{i_1 \% 2^{N'-t-1}}^{N'-t-1} + 2d_{i_2 \% 2^{N'-t-1}}^{N'-t-1} - d_{i_3 \% 2^{N'-t-1}}^{N'-t-1}) / 2^{k-t}$$

$$k \leq t :$$

$$d_{i_{new}}^{N'-k} = d_{(i+s/2^k) \% 2^{N'-k}}^{N'-k}$$

where,

$$i_1 = 2^{k-t-1}i + \lfloor s/2^{t+1} \rfloor$$

$$i_2 = 2^{k-t-2}(2i+1) + \lfloor s/2^{t+1} \rfloor$$

$$i_3 = 2^{k-t-1}(i+1) + \lfloor s/2^{t+1} \rfloor$$

and,

$$d_{i_2 \% 2^{N'-1}}^{N'-t-1} = 0, \text{ if } i_2 \text{ is a non-integer} \quad (2.25)$$

The worst case complexity of the above formula is  $O(\log(L + 2^h))$  (again please refer to the Complexity Analysis section for more details).

## 2.5 N-DIMENSIONAL SHIFT

Due to separability, an N-dimensional standard Haar transform is constructed by applying the one-dimensional transform along each dimension. As a result, the above solution can also be easily generalized to N-dimensional signals by applying it along each dimension separately.

## 2.6 COMPLEXITY ANALYSIS

In this section we explain in further detail the complexity of evaluating  $d_{i_{new}}^{N-k}$  using equation (2.13),  $A_{i_{new}}^{N-k}$  using equation (2.23) and  $d_{i_{new}}^{N-k}$  using equation (2.25).

By examining (2.13), it is easy to verify that the complexity of evaluating  $d_{i_{new}}^{N-k}$  can be expressed by the difference of the bounds of the two sums in the equation, that is  $O(i_3 - i_1)$ . Substituting the values for  $i_1$  and  $i_3$ , the complexity can be shown to be  $O(2^{k-t-1})$  when  $k > t$ . When  $k \leq t$  the complexity becomes  $O(1)$ . Therefore, one can determine that the worst case is when  $t = 0$ , that is when the shift is odd. In that case the complexity of computing  $d_{i_{new}}^{N-k}$  becomes  $O(2^{k-1})$ . Let

$L = 2^N$  be the size of the signal, then the number of the wavelet coefficients in a fully transformed signal is  $L - 1 = 2^N - 1$ . At reduction level  $k = N$ , i.e. the root, the complexity of evaluating  $d_{0_{new}}^0$  is  $O(2^{N-1}) = O(\frac{L}{2})$  with a probability of  $\frac{1}{L-1}$ . At the next reduction level  $k = N - 1$ , the complexity is  $O(2^{(N-1)-1}) = O(\frac{L}{2^2})$  with a probability of  $\frac{2}{L-1}$ . Table (2.6) shows the complexity and its probability at each reduction level  $k$ .

Table 2.1: Table of the complexity and probability at each reduction level  $k$  for the one-dimensional wavelet coefficient  $d_{i_{new}}^{N-k}$ .

Reduction Level	Complexity	Probability= $\frac{\text{Number of Coefficients at } k}{\text{Number of Coefficients}}$
$k = N$	$O(\frac{L}{2})$	$\frac{1}{L-1}$
$k = N - 1$	$O(\frac{L}{2^2})$	$\frac{2}{L-1}$
$k = N - 2$	$O(\frac{L}{2^3})$	$\frac{2^2}{L-1}$
$k = N - 3$	$O(\frac{L}{2^4})$	$\frac{2^3}{L-1}$
$\vdots$	$\vdots$	$\vdots$
$k = 1$	$O(\frac{L}{2^N})$	$\frac{2^{N-1}}{L-1}$

By multiplying the complexities and the probabilities in table (2.6) and summing them up, the average performance of the worst case for evaluating  $d_{i_{new}}^{N-k}$  is found to be  $O(\log(L))$ .

By following a similar analysis and examining (2.23), one can find that the complexity for evaluating  $A_{i_{new}}^{N-k}$  is  $O(\log(L))$  as well. Also, by examining (2.25) one can find that complexity for

evaluating  $d_{i_{new}}^{N-k}$  after a non-integer shift is  $O(\log(L + 2^h))$ , where  $h$  is the number of levels added to achieve the shift.

## 2.7 EXPERIMENTAL VALIDATION

We validate our results on the problem of accurate image rotation using the decomposition of the rotation matrix described in [KG85, Pae86, DH92, UTY95]. The choice of this application is driven by the fact that it allows us to evaluate all aspects such as integer and non-integer shifts, and the separability property.

### 2.7.1 IMAGE ROTATION

We implement rotation as a sequence of sheers using the following factorization [KG85, Pae86, DH92, UTY95]:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} = \begin{bmatrix} 1 & -\tan(\frac{\theta}{2}) \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ \sin(\theta) & 1 \end{bmatrix} \times \begin{bmatrix} 1 & -\tan(\frac{\theta}{2}) \\ 0 & 1 \end{bmatrix} \quad (2.26)$$













A shear is in fact a sequence of shifts that are row-dependent, if the shear is horizontal, and column-dependent if it is vertical. That is, each row is shifted by  $\Delta x = -y \cdot \tan \frac{\theta}{2}$  in a horizontal shear while each column is shifted by  $\Delta y = x \cdot \sin \theta$  in a vertical shear. Note that  $\Delta x$  and  $\Delta y$  are in general non-integer values, hence, the applicability of our phase-shifting relations derived in the previous sections. Figure (2.2-b) shows the application of our method to the 3-step shearing image rotation with  $h = 3$ . Figure (2.3) shows a magnified portion of the image under different  $h$  values. An integer shift ( $h = 0$ ) results in a jagged effect. This effect is eliminated, leading to higher quality results, as we increase the value of  $h$ . Note that visually satisfactory results are obtained even with  $h = 2$ .

As noted in [UTY95], the worst scenario occurs when the errors get accumulated. Therefore, in order to quantify the performance, we computed the residual error, using an experiment similar to the one adopted in [UTY95]. In other words, we successively rotated an input image by  $\frac{\pi}{8}$  until it rotated back to its original position. Figures 2.4 and 2.5 show the results and the associated residual errors on two standard test images for our method as compared to the nearest-neighbor, bilinear, bicubic, and the sinc method. Note that the image in Figure 2.4, which was also used by [UTY95], is specifically designed for capturing accumulated errors in successive rotations. We tested and compared our method extensively on many images, some of which are shown in table 3.7.



Figure 2.2: Original image, and the rotated one by 45 degrees using (2.25) with  $h = 3$ .

Table 2.2: Quantification and comparison of the accumulated residual error on several test images.

										
Nearest Neighbor	23.2451	15.3117	23.2249	19.3687	26.9128	13.8441	22.1845	7.2702	11.8229	18.3561
Bilinear	21.9343	12.7582	21.9399	18.3487	26.5292	12.0671	20.0879	6.2334	10.3405	17.02
Bicubic	15.2645	7.0842	14.5404	11.4509	17.2482	6.7327	11.6671	4.7183	5.9234	9.6489
Sinc	8.4349	1.8098	4.7284	4.0193	6.4774	2.3743	2.9468	2.2373	2.042	2.4533
Our Method	3.3738	1.5586	3.0092	2.4095	3.4965	1.5173	2.3753	1.1139	1.3243	2.0574

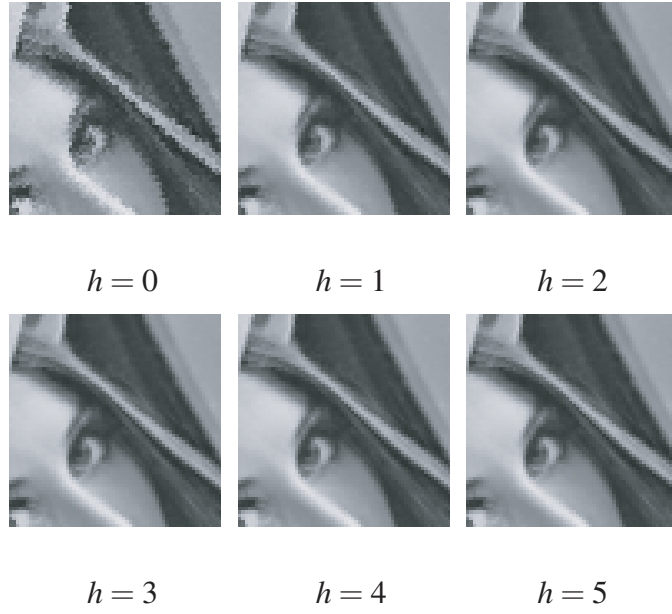


Figure 2.3: A magnified portion of the image rotated using equation (2.25). a. integer shift. b. non-integer shift with precision of  $\frac{1}{2}$ . c. non-integer shift with precision of  $\frac{1}{2^2}$ . d. non-integer shift with precision of  $\frac{1}{2^3}$ . e. non-integer shift with precision of  $\frac{1}{2^4}$ . f. non-integer shift with precision of  $\frac{1}{2^5}$ .

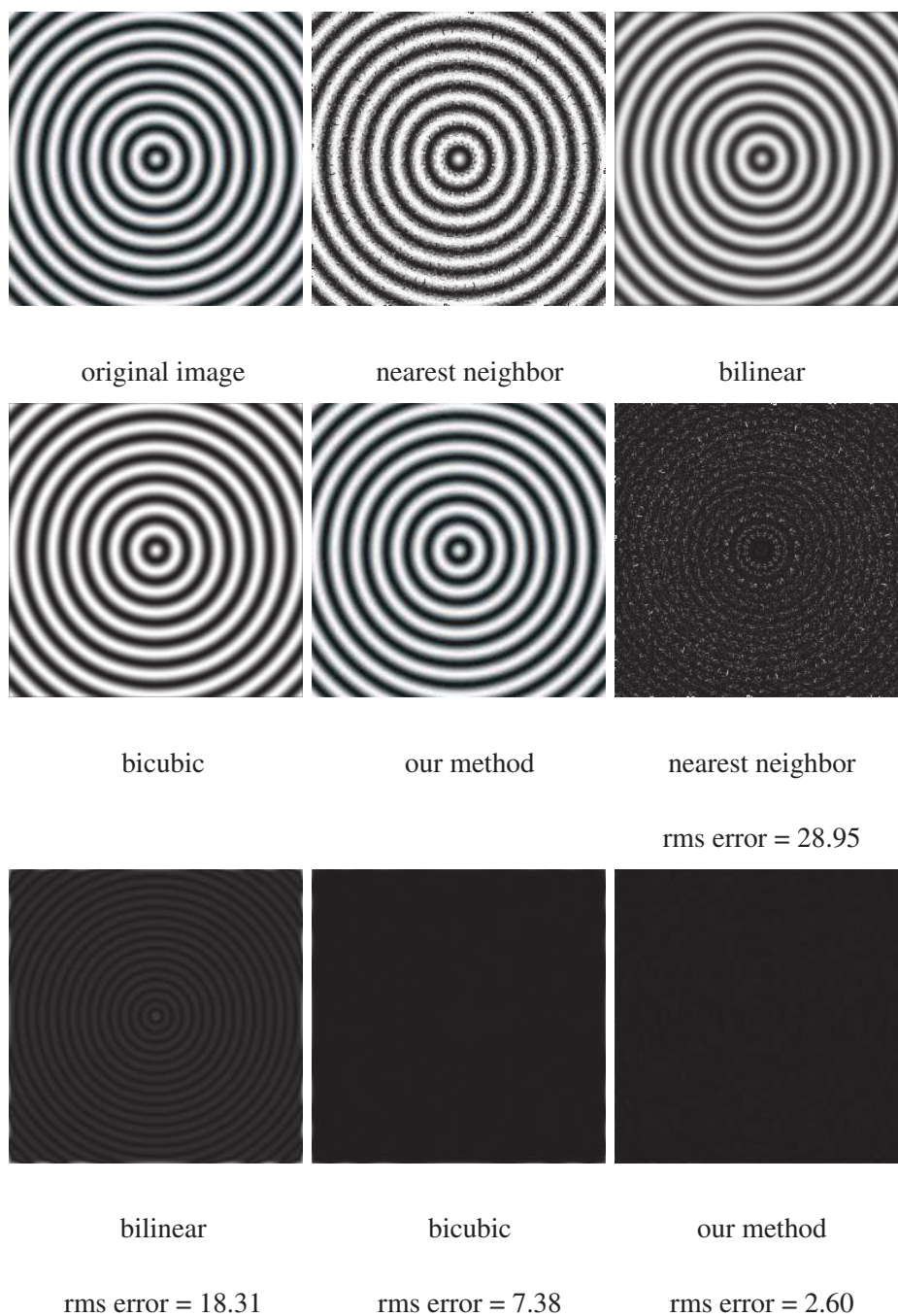


Figure 2.4: The above images show the results of successively rotating the original image 16 times by a degree of  $\frac{\pi}{8}$  for different methods including ours

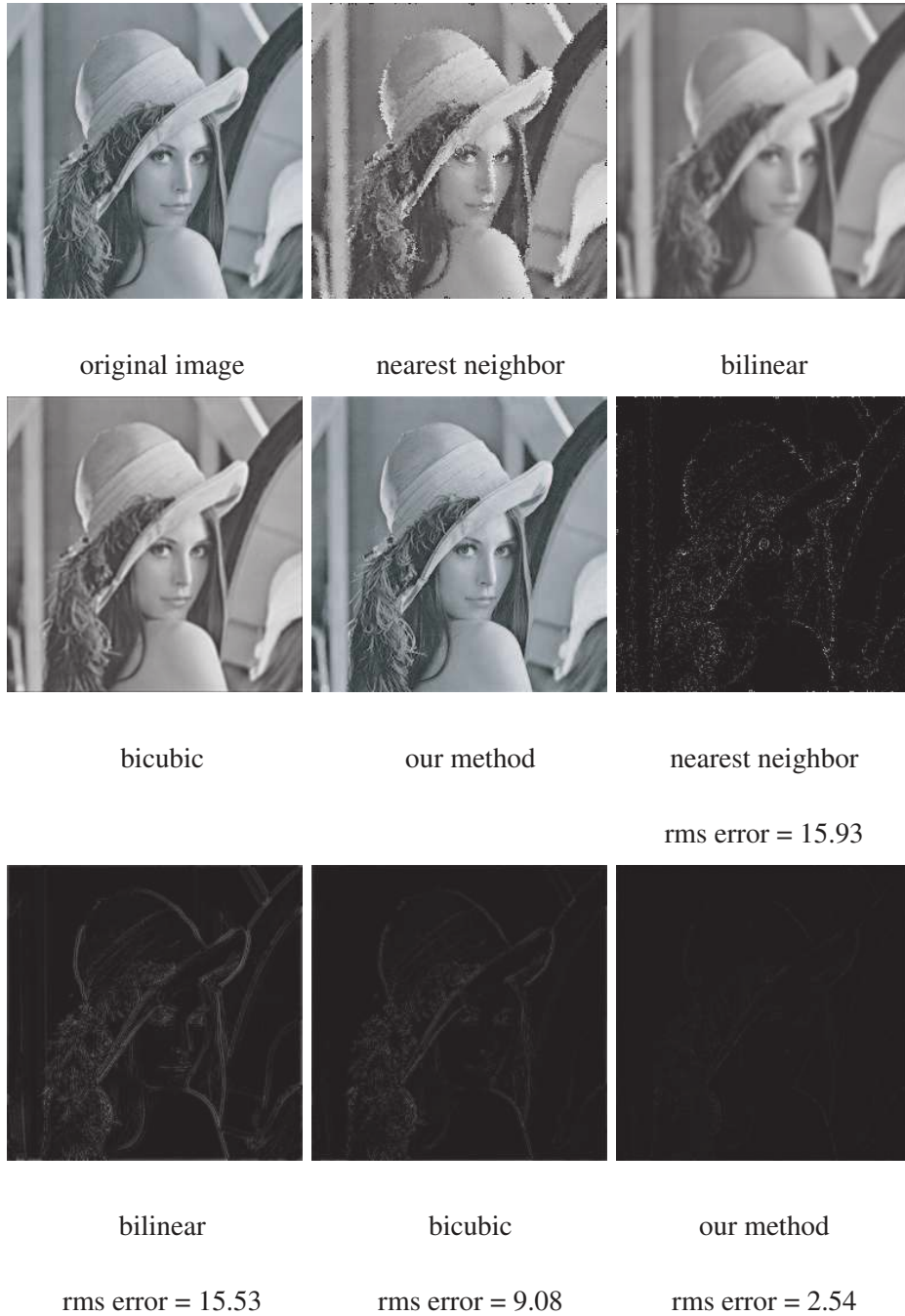


Figure 2.5: The above images show the results of successively rotating the original image 16 times by a degree of  $\frac{\pi}{8}$  for different methods including ours

### 3 PHASE-SHIFTING IN THE NON-SEPARABLE HAAR DOMAIN

#### 3.1 NOTATIONS AND SETUP

Let  $x(n, m)$  be a two-dimensional signal of size  $2^N \times 2^N$ , where  $N$  is a positive integer. The Haar transform of  $x(n, m)$ , namely  $H(x(n, m))$ , has the form shown in figure (3.1).

such that  $A_{0,0}^0$  is the dc value of the signal and  $a_{i,j}^l$ ,  $b_{i,j}^l$  and  $c_{i,j}^l$  are the  $ij^{th}$  horizontal, vertical and diagonal wavelet coefficients respectively at level  $l$ , where  $l = 0, \dots, N-1$  for the 0-shift shift,  $i = 0, \dots, 2^l - 1$  and  $j = 0, \dots, 2^l - 1$ .

Transforming a signal using Haar wavelets can be expressed using a tree as in fig (3.2). The tree is constructed of  $N$  levels with  $x(n, m)$  residing at the leaves, i.e. the  $N^{th}$  level. The  $ij^{th}$  node at level  $l$  in the tree is made to hold the  $ij^{th}$  scaling coefficient  $A_{i,j}^l$  and the wavelet coefficients  $a_{i,j}^N$ ,  $b_{i,j}^N$  and  $c_{i,j}^N$ , where  $l = 0, \dots, N-1$ ,  $i = 0, \dots, 2^l - 1$  and  $j = 0, \dots, 2^l - 1$ . We denote  $x(i, j)$  as  $A_{i,j}^N$  and let  $l = 0, \dots, N$  with  $a_{i,j}^N = 0$ ,  $b_{i,j}^N = 0$  and  $c_{i,j}^N = 0$ . For brevity, we will sometimes refer to the horizontal, vertical and diagonal wavelet coefficients by  $a$ ,  $b$  and  $c$ , respectively.

$\mathbf{A}_{0,0}^0$	$\mathbf{a}_{0,0}^0$	$\mathbf{a}_{0,0}^1$	$\mathbf{a}_{0,1}^1$	...	$\mathbf{a}_{0,0}^{N-1}$	...	$\mathbf{a}_{0,2^{N-1}-1}^{N-1}$
$\mathbf{b}_{0,0}^0$	$\mathbf{c}_{0,0}^0$	$\mathbf{a}_{1,0}^1$	$\mathbf{a}_{1,1}^1$	...	$\mathbf{a}_{1,0}^{N-1}$	...	$\mathbf{a}_{1,2^{N-1}-1}^{N-1}$
$\mathbf{b}_{0,1}^1$	$\mathbf{b}_{0,1}^1$	$\mathbf{c}_{0,0}^1$	$\mathbf{c}_{0,1}^1$	...	$\mathbf{a}_{2,0}^{N-1}$	...	$\mathbf{a}_{2,2^{N-1}-1}^{N-1}$
$\mathbf{b}_{1,0}^1$	$\mathbf{b}_{1,1}^1$	$\mathbf{c}_{1,0}^1$	$\mathbf{c}_{1,1}^1$	...	...	...	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\mathbf{a}_{2^{N-1}-1,0}^{N-1}$	...	$\mathbf{a}_{2^{N-1}-1,2^{N-1}-1}^{N-1}$
$\mathbf{b}_{0,0}^{N-1}$	$\mathbf{b}_{0,1}^{N-1}$	$\mathbf{b}_{0,2}^{N-1}$	...	$\mathbf{b}_{0,2^{N-1}-1}^{N-1}$	$\mathbf{c}_{0,0}^{N-1}$	...	$\mathbf{c}_{0,2^{N-1}-1}^{N-1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	...	...	...
$\mathbf{b}_{2^{N-1}-1,0}^{N-1}$	$\mathbf{b}_{0,0}^{N-1}$	$\mathbf{b}_{0,0}^{N-1}$	...	$\mathbf{b}_{2^{N-1}-1,2^{N-1}-1}^{N-1}$	$\mathbf{c}_{2^{N-1}-1,0}^{N-1}$	...	$\mathbf{c}_{2^{N-1}-1,2^{N-1}-1}^{N-1}$

Figure 3.1: Two dimensional Haar transform

Let

$$\begin{aligned}
X_{i,j}^l &= a_{[i/2],[j/2]}^{l-1} + b_{[i/2],[j/2]}^{l-1} + c_{[i/2],[j/2]}^{l-1} \\
Y_{i,j}^l &= -a_{[i/2],[j/2]}^{l-1} + b_{[i/2],[j/2]}^{l-1} - c_{[i/2],[j/2]}^{l-1} \\
Z_{i,j}^l &= a_{[i/2],[j/2]}^{l-1} - b_{[i/2],[j/2]}^{l-1} - c_{[i/2],[j/2]}^{l-1} \\
W_{i,j}^l &= -a_{[i/2],[j/2]}^{l-1} - b_{[i/2],[j/2]}^{l-1} + c_{[i/2],[j/2]}^{l-1}
\end{aligned} \tag{3.1}$$

The following formula shows the relation between the scaling coefficient  $A_{i,j}^l$  and its parent at level

$l-1$ :

$$A_{i,j}^l = \begin{cases} A_{i/2,j/2}^{l-1} + X_{i/2,j/2}^l, & i \text{ is even, } j \text{ is even} \\ A_{i/2,[j/2]}^{l-1} + Y_{i/2,[j/2]}^l, & i \text{ is even, } j \text{ is odd} \\ A_{[i/2],j/2}^{l-1} + Z_{[i/2],j/2}^l, & i \text{ is odd, } j \text{ is even} \\ A_{[i/2],[j/2]}^{l-1} + W_{[i/2],[j/2]}^l, & i \text{ is odd, } j \text{ is odd} \end{cases} \tag{3.2}$$

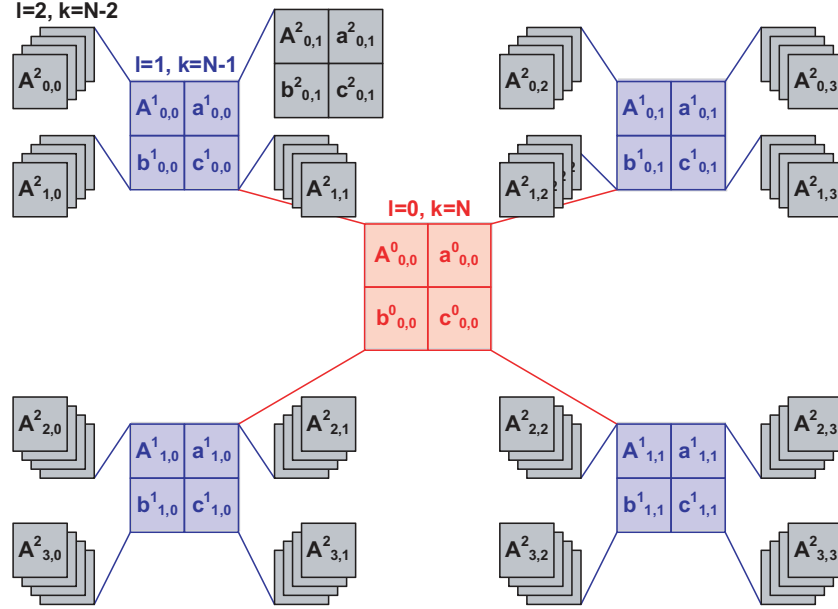


Figure 3.2: The Haar transform of the two-dimensional signal  $x(m,n)$  is composed of the dc value  $A^0_{0,0}$  and the wavelet coefficients  $a^l_{i,j}$ ,  $b^l_{i,j}$  and  $c^l_{i,j}$ , where  $l = 0, \dots, 2^{N-1}$ ,  $i = 0, \dots, 2^l - 1$  and  $j = 0, \dots, 2^l - 1$ . The scaling coefficients  $A^l_{i,j}$  at each level are used to help derive the equation for phase-shifting the signal  $x(m,n)$ , but are not used in the final form of the equation. Only the first three levels of the tree are shown with one  $l$  to avoid cluttering the figure.

We let  $D^l_{i,j}$  be the difference between  $A^0_{0,0}$  and  $A^l_{i,j}$ , then

$$A^l_{i,j} = A^0_{0,0} + D^l_{i,j} \quad (3.3)$$



By substituting (3.3) into (3.2),  $D_{i,j}^l$  can be computed recursively solely in terms of the wavelet coefficients as follows:

$$D_{i,j}^l = \begin{cases} D_{i/2,j/2}^{l-1} + X_{i/2,j/2}^l, & i \text{ is even, } j \text{ is even} \\ D_{i/2,\lfloor j/2 \rfloor}^{l-1} + Y_{i/2,\lfloor j/2 \rfloor}^l, & i \text{ is even, } j \text{ is odd} \\ D_{\lfloor i/2 \rfloor,j/2}^{l-1} + Z_{\lfloor i/2 \rfloor,j/2}^l, & i \text{ is odd, } j \text{ is even} \\ D_{\lfloor i/2 \rfloor,\lfloor j/2 \rfloor}^{l-1} + W_{\lfloor i/2 \rfloor,\lfloor j/2 \rfloor}^l, & i \text{ is odd, } j \text{ is odd} \\ 0, & i = j = l = 0 \end{cases} \quad (3.4)$$

A two-dimensional signal can be shifted horizontally or vertically. Both types of shifts affect the  $a_{i,j}^l$ ,  $b_{i,j}^l$  and  $c_{i,j}^l$  at all levels. At level  $N - k$ , there are  $2^k \times 2^k$  non-redundant coefficient sets each of size  $2^{N-k} \times 2^{N-k}$  [SB97], where  $k = 1, \dots, N$ . A horizontal shift  $s_h = 0, \dots, 2^N - 1$  or a vertical shift  $s_v = 0, \dots, 2^N - 1$  can be one of the following possibilities:

- A shift that is divisible by  $2^k$ .
- An odd shift.
- An even shift that is not divisible by  $2^k$ .

We derive the formulae for evaluating the  $a$  wavelet coefficients under a horizontal shift  $s_h$ , for each of the three possibilities, followed by similar derivations for  $b$  and  $c$  wavelet coefficients. Formulae for vertical shifts can be derived in a similar manner.

## 3.2 HORIZONTAL COEFFICIENTS FOR HORIZONTAL SHIFT

### 3.2.1 SHIFTING BY A MULTIPLE OF $2^k$

This is the simplest case. A shift  $s_h$  in the discrete time domain that is equal to  $2^k u$  is a horizontal circular shift of the 0-shift wavelet coefficients at level  $N - k$  by  $u$ , that is,

$$a_{i,j_{new}}^{N-k} = a_{i,(j+u)\%2^{N-k}}^{N-k}, \quad k = 1, \dots, N \quad (3.5)$$

where  $0 \leq u \leq 2^{N-k} - 1$  and  $\%$  is the mod operation. Notice that for levels  $N - (k - 1), N - (k - 2), \dots, N - 1$  a horizontal shift of  $2^k u$  in the time domain is a horizontal circular shift of the coefficients at those levels by  $2u, 2^2 u, \dots, 2^{k-1} u$  respectively. In other words, a horizontal shift of  $2^k u$  in the time domain shifts the coefficients at level  $N - k$  horizontally by  $u$ , while shifting the coefficients at level  $N - (k - 1)$  horizontally by twice as much, and the coefficients at level  $N - (k - 2)$  by four times as much and so on.

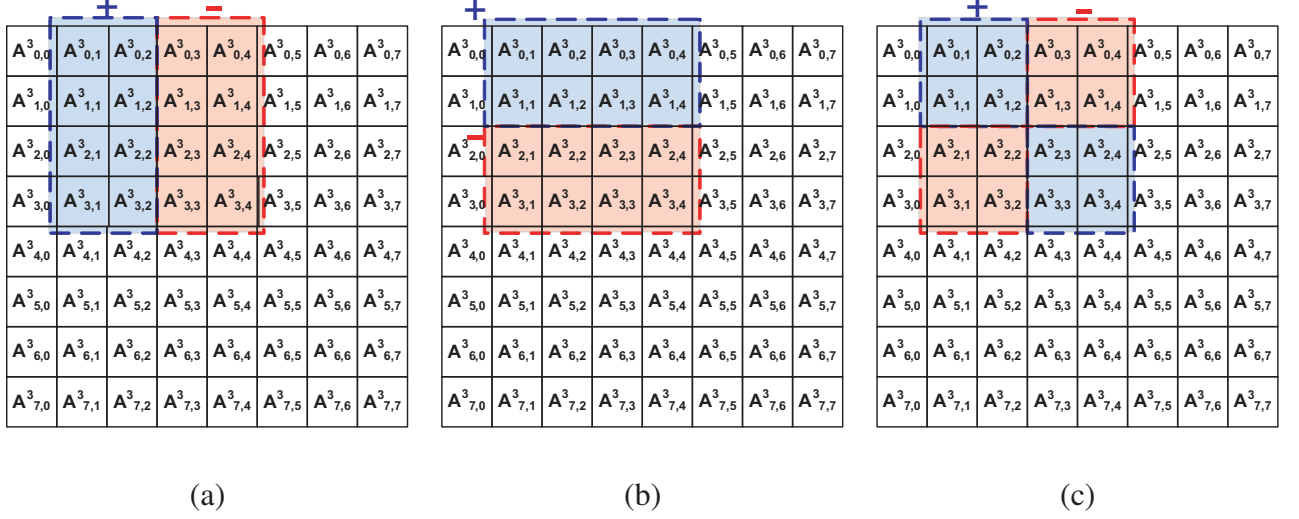


Figure 3.3: The above figures show the scaling coefficients at level  $l = 3$ . In (a), (b), and (c), respectively, the rectangular windows show the coefficients included in evaluating  $a_{0,0_{new}}^2$ ,  $b_{0,0_{new}}^2$ , and  $c_{0,0_{new}}^2$  under a horizontal odd shift of one. These are evaluated by summing and subtracting the highlighted quadrants in the windows as shown in the diagrams.

### 3.2.2 SHIFTING BY AN ODD AMOUNT

By examining the tree in Fig. 3.2, we notice that:

$$a_{i,j_{new}}^{N-k} = \left( \sum_{m=2^k i}^{2^{k(i+1)}} ((A_{m,j_1 \% 2^N}^N + \dots + A_{m,(j_2-1) \% 2^N}^N) - (A_{m,j_2 \% 2^N}^N + \dots + A_{m,(j_3-1) \% 2^N}^N)) \right) / 4^k$$

where,

$$j_1 = 2^k j + s_h$$

$$j_2 = 2^{k-1}(2j+1) + s_h$$

$$j_3 = 2^k(j+1) + s_h \tag{3.6}$$

The above equation evaluates  $a_{i,j_{new}}^{N-k}$  by summing the first half and subtracting the second half of each row of scaling coefficients at the leaves level  $N$ , which fall inside the window determined by the two corners  $(2^k i, 2^k j + s_h)$  and  $(2^k(i+1), 2^k(j+1) + s_h)$ . Figure 3.3-(a) illustrates the evaluation of  $a_{0,0}^2$  using the scaling coefficients at level  $l = 3$  under a horizontal odd shift of one.

Substituting (3.3), (3.4) and then (3.1) into (3.6), we get the coefficients of the shifted signal as follows:

$$\begin{aligned}
a_{i,j_{new}}^{N-k} = & \left( \sum_{m=2^{k-1}i}^{2^{k-1}(i+1)-1} (D_{m,j_1 \% 2^{N-1}}^{N-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m,n \% 2^{N-1}}^{N-1} \right. \\
& - 2 \sum_{n=j_2+1}^{j_3-1} D_{m,n \% 2^{N-1}}^{N-1} - D_{m,j_3 \% 2^{N-1}}^{N-1} \\
& \left. - a_{m,j_1 \% 2^{N-1}}^{N-1} + 2a_{m,j_2 \% 2^{N-1}}^{N-1} - a_{m,j_3 \% 2^{N-1}}^{N-1} \right) / 2^{2k-1}
\end{aligned}$$

where,

$$\begin{aligned}
j_1 &= 2^{k-1}j + \lfloor s_h/2 \rfloor \\
j_2 &= 2^{k-2}(2j+1) + \lfloor s_h/2 \rfloor \\
j_3 &= 2^{k-1}(j+1) + \lfloor s_h/2 \rfloor
\end{aligned} \tag{3.7}$$

Note that at  $k = 1$ ,  $j_2$  is a non-integer value. When that is the case we set  $a_{i,j_2 \% 2^{N-1}}^{N-1}$  to 0.

### 3.2.3 SHIFTING BY AN EVEN AMOUNT THAT IS NOT DIVISIBLE BY $2^k$

In this case,  $s_h$  is divisible by  $2^t$ , where  $1 \leq t \leq k-1$  and  $2^t$  is the highest power of 2 by which  $s_h$  is divisible. This allows us to let  $s_h = 2^t u$ , where  $0 \leq u \leq 2^{N-t} - 1$ , which means that the coefficients at levels  $N-1, \dots, N-t$  follow the first case. In other words, the 0-shift coefficients at levels  $N-1, N-2, \dots, N-t$  are circularly shifted in the horizontal direction by  $2^{t-1}u, 2^{t-2}u, \dots, u$ , respectively. On the other hand,  $a_{i,j_{new}}^{N-k}$  is verified to be an odd shift of the scaling coefficients at level  $N-t$ . In other words, at level  $N-k$ ,  $a_{i,j_{new}}^{N-k}$  can be evaluated using the following modification of equation (3.6):

$$a_{i,j_{new}}^{N-k} = \left( \sum_{m=2^{k-t}i}^{2^{k-t}(i+1)} ((A_{m,j_1 \% 2^{N-t}}^{N-t} + \dots + A_{m,(j_2-1) \% 2^{N-t}}^{N-t}) - (A_{m,j_2 \% 2^{N-t}}^{N-t} + \dots + A_{m,(j_3-1) \% 2^{N-t}}^{N-t})) \right) / 4^{k-t}$$

where,

$$\begin{aligned} j_1 &= 2^{k-t}j + s_h/2^t \\ j_2 &= 2^{k-t-1}(2j+1) + s_h/2^t \\ j_3 &= 2^{k-t}(j+1) + s_h/2^t \end{aligned} \tag{3.8}$$

Following the same steps as above, we get:

$$\begin{aligned}
a_{i,j_{new}}^{N-k} &= \left( \sum_{m=2^{k-t-1}i}^{2^{k-t-1}(i+1)-1} (D_{m,j_1 \% 2^{N-t-1}}^{N-t-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m,n \% 2^{N-t-1}}^{N-t-1} \right. \\
&\quad - 2 \sum_{n=j_2+1}^{j_3-1} D_{m,n \% 2^{N-t-1}}^{N-t-1} - D_{m,j_3 \% 2^{N-t-1}}^{N-t-1} \\
&\quad \left. - a_{m,j_1 \% 2^{N-t-1}}^{N-t-1} + 2a_{m,j_2 \% 2^{N-t-1}}^{N-t-1} - a_{m,j_3 \% 2^{N-t-1}}^{N-t-1} \right) / 2^{2k-2t-1}
\end{aligned}$$

where,

$$\begin{aligned}
j_1 &= 2^{k-t-1}j + \lfloor s_h/2^{t+1} \rfloor \\
j_2 &= 2^{k-t-2}(2j+1) + \lfloor s_h/2^{t+1} \rfloor \\
j_3 &= 2^{k-t-1}(j+1) + \lfloor s_h/2^{t+1} \rfloor
\end{aligned} \tag{3.9}$$

Note that the second case is the same as the third case where  $t = 0$ . This gives rise to the following final formula for evaluating the  $a$  wavelet coefficients under a horizontal shift:

$k > t :$

$$\begin{aligned} a_{i,j_{new}}^{N-k} &= \left( \sum_{m=2^{k-t-1}i}^{2^{k-t-1}(i+1)-1} (D_{m,j_1 \% 2^{N-t-1}}^{N-t-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m,n \% 2^{N-t-1}}^{N-t-1} \right. \\ &\quad - 2 \sum_{n=j_2+1}^{j_3-1} D_{m,n \% 2^{N-t-1}}^{N-t-1} - D_{m,j_3 \% 2^{N-t-1}}^{N-t-1} \\ &\quad \left. - a_{m,j_1 \% 2^{N-t-1}}^{N-t-1} + 2a_{m,j_2 \% 2^{N-t-1}}^{N-t-1} - a_{m,j_3 \% 2^{N-t-1}}^{N-t-1} \right) / 2^{2k-2t-1} \end{aligned}$$

$k \leq t :$

$$a_{i,j_{new}}^{N-k} = a_{i,(j+s_h/2^k) \% 2^{N-k}}^{N-k}$$

where,

$$\begin{aligned} j_1 &= 2^{k-t-1}j + \lfloor s_h/2^{t+1} \rfloor \\ j_2 &= 2^{k-t-2}(2j+1) + \lfloor s_h/2^{t+1} \rfloor \\ j_3 &= 2^{k-t-1}(j+1) + \lfloor s_h/2^{t+1} \rfloor \end{aligned} \tag{3.10}$$

### 3.3 VERTICAL COEFFICIENTS FOR HORIZONTAL SHIFT

For this section and the next one, again we let  $s_h = 2^t u$  be a horizontal shift, where  $0 \leq s_h \leq 2^N - 1$ ,  $0 \leq t \leq N$ ,  $2^t$  is the highest power of 2 by which  $s_h$  is divisible, and  $0 \leq u \leq 2^{N-t} - 1$  is an odd

positive integer. Then for all  $k \leq t$ ,

$$b_{i,j_{new}}^{N-k} = b_{i,(j+s_h/2^k)\%2^{N-k}}^{N-k} \quad (3.11)$$

To compute the  $b_{i,j_{new}}^{N-k}$  coefficient for  $k > t$  after a horizontal shift, we write  $b_{i,j_{new}}^{N-k}$  in terms of the coefficients at level  $N - t$ :

$$\begin{aligned} b_{i,j_{new}}^{N-k} &= \left( \sum_{m=2^{k-t}i}^{2^{k-t-1}(2i+1)-1} (A_{m,j_1\%2^{N-t}}^{N-t} + \dots + A_{m,(j_2-1)\%2^{N-t}}^{N-t}) \right. \\ &\quad \left. - \sum_{m=2^{k-t-1}(2i+1)}^{2^{k-t}(i+1)-1} (A_{m,j_1\%2^{N-t}}^{N-t} + \dots + A_{m,(j_2-1)\%2^{N-t}}^{N-t}) \right) / 4^{k-t} \end{aligned}$$

where,

$$\begin{aligned} j_1 &= 2^{k-t}j + s_h/2^t \\ j_2 &= 2^{k-t}(j+1) + s_h/2^t \end{aligned} \quad (3.12)$$

The above equation defines a window that encloses the scaling coefficients at level  $N - t$  that fall between the two corners  $(2^{k-t}i, 2^{k-t}j + s_h/2^t)$  and  $(2^{k-t}(i+1), 2^{k-t}(j+1) + s_h/2^t)$ , which are the same bounds used to compute  $a_{i,j_{new}}^{N-k}$  in (3.10).  $b_{i,j_{new}}^{N-k}$  is found by summing each row of scaling coefficients in the upper half of the window and subtracting the sum of each row in the lower half. Figure 3.3-(b) illustrates the computation of  $b_{0,0}^2$  using the scaling coefficients at level  $l = 3$  under a horizontal odd shift of one.



By substituting (3.3), (3.4) and (3.1) in the above, and combining the result with (3.11) we get the coefficients of the shifted signal:

$k > t + 1 :$

$$\begin{aligned}
b_{i,j_{new}}^{N-k} = & \left( \sum_{m=2^{k-t-1}i}^{2^{k-t-2}(2i+1)-1} (D_{m,j_1 \% 2^{N-t-1}}^{N-t-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m,n \% 2^{N-t-1}}^{N-t-1} \right. \\
& + D_{m,j_2 \% 2^{N-t-1}}^{N-t-1} - a_{m,j_1 \% 2^{N-t-1}}^{N-t-1} + a_{m,j_2 \% 2^{N-t-1}}^{N-t-1}) \\
& - \sum_{m=2^{k-t-2}(2i+1)}^{2^{k-t-1}(i+1)-1} (D_{m,j_1 \% 2^{N-t-1}}^{N-t-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m,n \% 2^{N-t-1}}^{N-t-1} \\
& + D_{m,j_2 \% 2^{N-t-1}}^{N-t-1} - a_{m,j_1 \% 2^{N-t-1}}^{N-t-1} + a_{m,j_2 \% 2^{N-t-1}}^{N-t-1}) \Big) / 2^{2k-2t-1}
\end{aligned}$$

$k = t + 1 :$

$$\begin{aligned}
b_{i,j_{new}}^{N-k} = & (b_{i,(j+\lfloor s_h/2^k \rfloor) \% 2^{N-k}}^{N-k} - c_{i,(j+\lfloor s_h/2^k \rfloor) \% 2^{N-k}}^{N-k} \\
& + b_{i,(j+\lfloor s_h/2^k \rfloor + 1) \% 2^{N-k}}^{N-k} + c_{i,(j+\lfloor s_h/2^k \rfloor + 1) \% 2^{N-k}}^{N-k}) / 2
\end{aligned}$$

$k \leq t :$

$$b_{i,j_{new}}^{N-k} = b_{i,(j+s_h/2^k) \% 2^{N-k}}^{N-k}$$

where,

$$\begin{aligned}
j_1 &= 2^{k-t-1}j + \lfloor s_h/2^{t+1} \rfloor \\
j_2 &= 2^{k-t-1}(j+1) + \lfloor s_h/2^{t+1} \rfloor
\end{aligned} \tag{3.13}$$

### 3.4 DIAGONAL COEFFICIENTS FOR HORIZONTAL SHIFT

Similar to 3.5 and 3.11, we can apply the following equation when the horizontal shift  $s_h$  is divisible by  $2^k$ :

$$c_{i,j_{new}}^{N-k} = c_{i,(j+s_h/2^k)\%2^{N-k}}^{N-k} \quad (3.14)$$

To compute the  $c_{i,j_{new}}^{N-k}$  coefficient for  $k > t$  after a horizontal shift, we write  $c_{i,j_{new}}^{N-k}$  in terms of coefficients at level  $N - t$ :

$$\begin{aligned} c_{i,j_{new}}^{N-k} &= \left( \sum_{m=2^{k-t}i}^{2^{k-t-1}(i+1)-1} ((A_{m,j_1\%2^{N-t}}^{N-t} + \dots + A_{m,(j_2-1)\%2^{N-t}}^{N-t}) \right. \\ &\quad - (A_{m,j_2\%2^{N-t}}^{N-t} + \dots + A_{m,(j_3-1)\%2^{N-t}}^{N-t})) \\ &\quad - \sum_{m=2^{k-t-1}(i+1)}^{2^{k-t}(i+1)-1} ((A_{m,j_1\%2^{N-t}}^{N-t} + \dots + A_{m,(j_2-1)\%2^{N-t}}^{N-t}) \\ &\quad \left. - (A_{m,j_2\%2^{N-t}}^{N-t} + \dots + A_{m,(j_3-1)\%2^{N-t}}^{N-t})) \right) / 4^{k-t} \end{aligned}$$

where,

$$\begin{aligned} j_1 &= 2^{k-t}j + s_h/2^t \\ j_2 &= 2^{k-t-1}(2j+1) + s_h/2^t \\ j_3 &= 2^{k-t}(j+1) + s_h/2^t \end{aligned} \quad (3.15)$$

The above equation defines a window which has the same bounds used to compute  $a_{i,j_{new}}^{N-k}$  and  $b_{i,j_{new}}^{N-k}$  in (3.10) and (3.13), respectively. By examining the example in Fig. 3.3-(c), one can see that  $c_{i,j_{new}}^{N-k}$  is computed by summing the first half and subtracting the second half of each row in the upper

half of the window, while subtracting the first half and summing the second half of each row in the lower half of the window, hence, the upper equation.

By substituting (3.3), (3.4) and (3.1) in the above, and combining the result with (3.14), we get:

$k > t + 1 :$

$$\begin{aligned}
c_{i,j_{new}}^{N-k} &= \left( \sum_{m=2^{k-t-1}i}^{2^{k-t-2}(2i+1)-1} (D_{m,j_1 \% 2^{N-t-1}}^{N-t-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m,n \% 2^{N-t-1}}^{N-t-1} - 2 \sum_{n=j_2+1}^{j_3-1} D_{m,n \% 2^{N-t-1}}^{N-t-1} \right. \\
&\quad - D_{m,j_3 \% 2^{N-t-1}}^{N-t-1} - a_{m,j_1 \% 2^{N-t-1}}^{N-t-1} + 2a_{m,j_2 \% 2^{N-t-1}}^{N-t-1} - a_{m,j_3 \% 2^{N-t-1}}^{N-t-1} ) \\
&\quad - \sum_{m=2^{k-t-2}(2i+1)}^{2^{k-t-1}(i+1)-1} (D_{m,j_1 \% 2^{N-t-1}}^{N-t-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m,n \% 2^{N-t-1}}^{N-t-1} - 2 \sum_{n=j_2+1}^{j_3-1} D_{m,n \% 2^{N-t-1}}^{N-t-1} \\
&\quad \left. - D_{m,j_3 \% 2^{N-t-1}}^{N-t-1} - a_{m,j_1 \% 2^{N-t-1}}^{N-t-1} + 2a_{m,j_2 \% 2^{N-t-1}}^{N-t-1} - a_{m,j_3 \% 2^{N-t-1}}^{N-t-1} ) \right) / 2^{2k-2t-1}
\end{aligned}$$

$k = t + 1 :$

$$\begin{aligned}
c_{i,j_{new}}^{N-k} &= (b_{i,(j+\lfloor s_h/2^k \rfloor) \% 2^{N-k}}^{N-k} - c_{i,(j+\lfloor s_h/2^k \rfloor) \% 2^{N-k}}^{N-k} \\
&\quad - b_{i,(j+\lfloor s_h/2^k \rfloor + 1) \% 2^{N-k}}^{N-k} - c_{i,(j+\lfloor s_h/2^k \rfloor + 1) \% 2^{N-k}}^{N-k}) / 2
\end{aligned}$$

$k \leq t :$

$$c_{i,j_{new}}^{N-k} = c_{i,(j+s_h/2^k) \% 2^{N-k}}^{N-k}$$

where,

$$\begin{aligned}
j_1 &= 2^{k-t-1}j + \lfloor s_h/2^{t+1} \rfloor \\
j_2 &= 2^{k-t-2}(2j+1) + \lfloor s_h/2^{t+1} \rfloor \\
j_3 &= 2^{k-t-1}(j+1) + \lfloor s_h/2^{t+1} \rfloor
\end{aligned} \tag{3.16}$$

The relations (3.10), (3.13), and (3.16) can now be used to compute the new wavelet coefficients of the Haar transform at all different levels after any horizontal shift  $s_h = 0, \dots, 2^N - 1$  using only the 0-shift coefficients. The equations for a vertical shift  $s_v = 0, \dots, 2^N - 1$  can be derived following the same steps that we used for horizontal shifting. They turn out to look the same as the horizontal case after interchanging the  $a$ 's with  $b$ 's,  $i$ 's with  $j$ 's and  $m$ 's with  $n$ 's:

$k > t + 1 :$

$$\begin{aligned}
a_{i,j_{new}}^{N-k} = & \left( \sum_{n=2^{k-t-1}j}^{2^{k-t-2}(2j+1)-1} (D_{i_1 \% 2^{N-t-1},n}^{N-t-1} + 2 \sum_{m=i_1+1}^{i_2-1} D_{m \% 2^{N-t-1},n}^{N-t-1} \right. \\
& + D_{i_2 \% 2^{N-t-1},n}^{N-t-1} - b_{i_1 \% 2^{N-t-1},n}^{N-t-1} + b_{i_2 \% 2^{N-t-1},n}^{N-t-1}) \\
& - \sum_{n=2^{k-t-2}(2j+1)}^{2^{k-t-1}(j+1)-1} (D_{i_1 \% 2^{N-t-1},n}^{N-t-1} + 2 \sum_{m=i_1+1}^{i_2-1} D_{m \% 2^{N-t-1},n}^{N-t-1} \\
& + D_{i_2 \% 2^{N-t-1},n}^{N-t-1} - b_{i_1 \% 2^{N-t-1},n}^{N-t-1} + b_{i_2 \% 2^{N-t-1},n}^{N-t-1}) \Big) / 2^{2k-2t-1}
\end{aligned}$$

$k = t + 1 :$

$$\begin{aligned}
a_{i,j_{new}}^{N-k} = & (a_{(i+\lfloor s_v/2^k \rfloor) \% 2^{N-k},j}^{N-k} - c_{(i+\lfloor s_v/2^k \rfloor) \% 2^{N-k},j}^{N-k} \\
& + a_{(i+\lfloor s_v/2^k \rfloor + 1) \% 2^{N-k},j}^{N-k} + c_{(i+\lfloor s_v/2^k \rfloor + 1) \% 2^{N-k},j}^{N-k}) / 2
\end{aligned}$$

$k \leq t :$

$$a_{i,j_{new}}^{N-k} = a_{(i+s_v/2^k) \% 2^{N-k},j}^{N-k}$$

where,

$$\begin{aligned}
i_1 &= 2^{k-t-1}i + \lfloor s_v/2^{t+1} \rfloor \\
i_2 &= 2^{k-t-1}(i+1) + \lfloor s_v/2^{t+1} \rfloor
\end{aligned} \tag{3.17}$$

$k > t :$

$$\begin{aligned}
b_{i,j_{new}}^{N-k} &= \left( \sum_{n=2^{k-t-1}j}^{2^{k-t-1}(j+1)-1} (D_{i_1 \% 2^{N-t-1},n}^{N-t-1} + 2 \sum_{m=i_1+1}^{i_2-1} D_{m \% 2^{N-t-1},n}^{N-t-1} \right. \\
&\quad - 2 \sum_{m=i_2+1}^{i_3-1} D_{m \% 2^{N-t-1},n}^{N-t-1} - D_{i_3 \% 2^{N-t-1},n}^{N-t-1} \\
&\quad \left. - b_{i_1 \% 2^{N-t-1},n}^{N-t-1} + 2b_{i_2 \% 2^{N-t-1},n}^{N-t-1} - b_{i_3 \% 2^{N-t-1},n}^{N-t-1} \right) / 2^{2k-2t-1}
\end{aligned}$$

$k \leq t :$

$$b_{i,j_{new}}^{N-k} = b_{(j+s_v/2^k) \% 2^{N-k},i}^{N-k}$$

where,

$$\begin{aligned}
i_1 &= 2^{k-t-1}i + \lfloor s_v/2^{t+1} \rfloor \\
i_2 &= 2^{k-t-2}(2i+1) + \lfloor s_v/2^{t+1} \rfloor \\
i_3 &= 2^{k-t-1}(i+1) + \lfloor s_v/2^{t+1} \rfloor
\end{aligned} \tag{3.18}$$

$k > t + 1 :$

$$\begin{aligned}
c_{i,j_{new}}^{N-k} &= \left( \sum_{n=2^{k-t-1}j}^{2^{k-t-2}(2j+1)-1} (D_{i_1 \% 2^{N-t-1},n}^{N-t-1} + 2 \sum_{m=i_1+1}^{i_2-1} D_{m \% 2^{N-t-1},n}^{N-t-1} - 2 \sum_{m=i_2+1}^{i_3-1} D_{m \% 2^{N-t-1},n}^{N-t-1} \right. \\
&\quad \left. - D_{i_3 \% 2^{N-t-1},n}^{N-t-1} - b_{i_1 \% 2^{N-t-1},n}^{N-t-1} + 2b_{i_2 \% 2^{N-t-1},n}^{N-t-1} - b_{i_3 \% 2^{N-t-1},n}^{N-t-1} \right) \\
&\quad - \sum_{n=2^{k-t-2}(2j+1)}^{2^{k-t-1}(j+1)-1} (D_{i_1 \% 2^{N-t-1},n}^{N-t-1} + 2 \sum_{m=i_1+1}^{i_2-1} D_{m \% 2^{N-t-1},n}^{N-t-1} - 2 \sum_{m=i_2+1}^{i_3-1} D_{m \% 2^{N-t-1},n}^{N-t-1} \\
&\quad \left. - D_{i_3 \% 2^{N-t-1},n}^{N-t-1} - b_{i_1 \% 2^{N-t-1},n}^{N-t-1} + 2b_{i_2 \% 2^{N-t-1},n}^{N-t-1} - b_{i_3 \% 2^{N-t-1},n}^{N-t-1} \right) / 2^{2k-2t-1}
\end{aligned}$$

$k = t + 1 :$

$$\begin{aligned}
c_{i,j_{new}}^{N-k} &= (a_{(i+\lfloor s_v/2^k \rfloor) \% 2^{N-k},j}^{N-k} - c_{(i+\lfloor s_v/2^k \rfloor) \% 2^{N-k},j}^{N-k} \\
&\quad - a_{(i+\lfloor s_v/2^k \rfloor + 1) \% 2^{N-k},j}^{N-k} - c_{(i+\lfloor s_v/2^k \rfloor + 1) \% 2^{N-k},j}^{N-k}) / 2
\end{aligned}$$

$k \leq t :$

$$c_{i,j_{new}}^{N-k} = c_{(i+s_v/2^k) \% 2^{N-k},j}^{N-k}$$

where,

$$\begin{aligned}
i_1 &= 2^{k-t-1}i + \lfloor s_v/2^{t+1} \rfloor \\
i_2 &= 2^{k-t-2}(2i+1) + \lfloor s_v/2^{t+1} \rfloor \\
i_3 &= 2^{k-t-1}(i+1) + \lfloor s_v/2^{t+1} \rfloor
\end{aligned} \tag{3.19}$$

Figures (3.4) show the horizontally, vertically and diagonally shifted images after reconstruction.



horizontal shift

vertical shift

diagonal shift

Figure 3.4: The above images were phase-shifted in the transform domain and reconstructed to demonstrate the shift.

### 3.5 SHIFTING PARTIALLY-TRANSFORMED SIGNALS

Equations (3.10), (3.13) and (3.16) can be used to evaluate the wavelet coefficients of a partially transformed shifted signal as in the fully transformed signal case. In this section we show how to evaluate the scaling coefficients at reduction step  $k$  for a two-dimensional signal that has been decomposed  $k$  times and shifted horizontally by the integer amount  $s_h$  in the time domain. The formula for shifting the signal vertically can be derived using the same analysis. We provide that formula at the end of this section.

### 3.5.1 SHIFTING BY A MULTIPLE OF $2^k$

Similar to evaluating the wavelet coefficients for a fully-transformed 2D signal, a shift  $s_h$  in the discrete domain that is equal to  $2^k u$  is a circular shift of the 0-shift scaling coefficients at level  $N - k$  by  $u$ , that is,

$$A_{i,j_{new}}^{N-k} = A_{i,(j+u)\%2^{N-k}}^{N-k}, \quad k = 1, \dots, N-1 \quad (3.20)$$

where  $0 \leq u \leq 2^{N-k} - 1$ .

### 3.5.2 SHIFTING BY AN ODD AMOUNT

By examining the tree in Fig. 3.2, we notice that:

$$A_{i,j_{new}}^{N-k} = \left( \sum_{m=2^k i}^{2^k(i+1)} (A_{m,j_1\%2^N}^N + \dots + A_{m,(j_2-1)\%2^N}^N) \right) / 4^k$$

where,

$$\begin{aligned} j_1 &= 2^k j + s_h \\ j_2 &= 2^k(j+1) + s_h \end{aligned} \quad (3.21)$$

The above equation evaluates  $A_{i,j_{new}}^{N-k}$  by summing the values of each row of scaling coefficients, at the leaves level  $N$ , which fall inside the window determined by the two corners  $(2^k i, 2^k j + s_h)$  and  $(2^k(i+1), 2^k(j+1) + s_h)$ .



Substituting (3.2) in the above, we get

$$A_{i,j_{new}}^{N-k} = \left( \sum_{m=2^k i}^{2^k(i+1)} (A_{0,0}^0 + D_{m,j_1 \% 2^N}^N + \dots + A_{0,0}^0 + D_{m,(j_2-1) \% 2^N}^N) \right) / 4^k \quad (3.22)$$

The number of  $A_{0,0}^0$ 's is equal to the number of coefficients  $A_{i,j}^l$  being summed, which is equal to  $4^k$ . We factor out  $A_{0,0}^0$ :

$$A_{i,j_{new}}^{N-k} = A_{0,0}^0 + \left( \sum_{m=2^k i}^{2^k(i+1)} (D_{m,j_1 \% 2^N}^N + \dots + D_{m,(j_2-1) \% 2^N}^N) \right) / 4^k \quad (3.23)$$

Substituting (3.4) and simplifying, we get the analytic solution for evaluating  $A_{i,j_{new}}^{N-k}$  under an odd horizontal shift  $s_h$ :

$$\begin{aligned} A_{i,j_{new}}^{N-k} &= A_{0,0}^0 + \left( \sum_{m=2^{k-1} i}^{2^{k-1}(i+1)-1} (D_{m,j_1 \% 2^{N-1}}^{N-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m,n \% 2^{N-1}}^{N-1} + D_{m,j_2 \% 2^{N-1}}^{N-1} \right. \\ &\quad \left. - a_{m,j_1 \% 2^{N-1}}^{N-1} + a_{m,j_2 \% 2^{N-1}}^{N-1}) \right) / 2^{2k-1} \end{aligned}$$

where,

$$\begin{aligned} j_1 &= 2^{k-1} j + \lfloor s_h / 2 \rfloor \\ j_2 &= 2^{k-1} (j+1) + \lfloor s_h / 2 \rfloor \end{aligned} \quad (3.24)$$

### 3.5.3 SHIFTING BY AN EVEN AMOUNT THAT IS NOT DIVISIBLE BY $2^k$

For a shift  $s_h = 2^t u$ , where  $0 \leq u \leq 2^{N-t} - 1$  and  $t < k$ , we can treat this case as an odd shift of the coefficients at level  $N - t$ , which is similar to what we did in evaluating the wavelet coefficients

under a shift  $s_h = 2^t u$ .  $A_{i,j_{new}}^{N-k}$  can now be evaluated using the following equation:

$$A_{i,j_{new}}^{N-k} = \left( \sum_{m=2^k i}^{2^k(i+1)} (A_{m,j_1 \% 2^{N-t}}^{N-t} + \dots + A_{m,(j_2-1) \% 2^{N-t}}^{N-t}) \right) / 4^k$$

where,

$$j_1 = 2^k j + s_h$$

$$j_2 = 2^k(j+1) + s_h \quad (3.25)$$

Proceeding as we did in the odd shift case, we get the following solution:

$$\begin{aligned} A_{i,j_{new}}^{N-k} &= A_{0,0}^0 + \left( \sum_{m=2^{k-1}i}^{2^{k-1}(i+1)-1} (D_{m,j_1 \% 2^{N-t-1}}^{N-t-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m,n \% 2^{N-t-1}}^{N-t-1} + D_{m,j_2 \% 2^{N-t-1}}^{N-t-1} \right. \\ &\quad \left. - a_{m,j_1 \% 2^{N-1}}^{N-t-1} + a_{m,j_2 \% 2^{N-t-1}}^{N-t-1}) \right) / 2^{2k-2t-1} \end{aligned}$$

where,

$$j_1 = 2^{k-1}j + \lfloor s_h/2 \rfloor$$

$$j_2 = 2^{k-1}(j+1) + \lfloor s_h/2 \rfloor \quad (3.26)$$

Combining the three cases, the final result becomes:

$k > t :$

$$A_{i,j_{new}}^{N-k} = A_{0,0}^0 + \left( \sum_{m=2^{k-1}i}^{2^{k-1}(i+1)-1} (D_{m,j_1 \% 2^{N-t-1}}^{N-t-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m,n \% 2^{N-t-1}}^{N-t-1} + D_{m,j_2 \% 2^{N-t-1}}^{N-t-1} - a_{m,j_1 \% 2^{N-1}}^{N-t-1} + a_{m,j_2 \% 2^{N-t-1}}^{N-t-1}) \right) / 2^{2k-2t-1}$$

$k \leq t :$

$$A_{i,j_{new}}^{N-k} = A_{i,(j+s_h/2^k) \% 2^{N-k}}^{N-k}$$

where,

$$j_1 = 2^{k-1}j + \lfloor s_h/2 \rfloor$$

$$j_2 = 2^{k-1}(j+1) + \lfloor s_h/2 \rfloor \quad (3.27)$$

Similarly, scaling coefficients  $A_{i,j}^{N-k}$  after a vertical shift  $s_v$  can be found to be the following:

$k > t :$

$$A_{i,j_{new}}^{N-k} = A_{0,0}^0 + \left( \sum_{n=2^{k-1}j}^{2^{k-1}(j+1)-1} (D_{i_1 \% 2^{N-t-1},n}^{N-t-1} + 2 \sum_{m=i_1+1}^{i_2-1} D_{m \% 2^{N-t-1},n}^{N-t-1} + D_{i_2 \% 2^{N-t-1},n}^{N-t-1} - b_{i_1 \% 2^{N-1},n}^{N-t-1} + b_{i_2 \% 2^{N-t-1},n}^{N-t-1}) \right) / 2^{2k-2t-1}$$

$k \leq t :$

$$A_{i,j_{new}}^{N-k} = A_{(i+s_v/2^k) \% 2^{N-k},j}^{N-k}$$

where,

$$i_1 = 2^{k-1}i + \lfloor s_v/2 \rfloor$$

$$i_2 = 2^{k-1}(i+1) + \lfloor s_v/2 \rfloor \quad (3.28)$$

The above relation can now be used to evaluate the new scaling coefficients of a partially transformed signal with  $k$  reduction steps after any horizontal shift  $s_h = 0, \dots, 2^N - 1$  or vertical shift  $s_v = 0, \dots, 2^N - 1$  using only the coefficients of the 0-shift signal. The worst case complexity for evaluating  $A_{i,j_{new}}^{N-k}$  using (3.27) is  $O(\log(L))$ , where  $L \times L$  is the size of the signal  $x(n, m)$  (see the Experimental Results and Discussion section for more details).

### 3.6 SUBPIXEL SHIFTING

Let the size of the signal be  $2^N \times 2^N$ ,  $N' = N + h$  and  $k = 1 + h, \dots, N + h$ , where  $h$  is the number of added levels. Equations (3.10), (3.13) and (3.13) can now be modified to allow for non-integer shifting by a precision of  $\frac{1}{2^h}$  by substituting  $N'$  for each  $N$  in the equations.

$k > t :$

$$\begin{aligned}
a_{i,j_{new}}^{N'-k} &= \left( \sum_{m=2^{k-t-1}i}^{2^{k-t-1}(i+1)-1} (D_{m,j_1 \% 2^{N'-t-1}}^{N'-t-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m,n \% 2^{N'-t-1}}^{N'-t-1} \right. \\
&\quad - 2 \sum_{n=j_2+1}^{j_3-1} D_{m,n \% 2^{N'-t-1}}^{N'-t-1} - D_{m,j_3 \% 2^{N'-t-1}}^{N'-t-1} \\
&\quad \left. - a_{m,j_1 \% 2^{N'-t-1}}^{N'-t-1} + 2a_{m,j_2 \% 2^{N'-t-1}}^{N'-t-1} - a_{m,j_3 \% 2^{N'-t-1}}^{N'-t-1} \right) / 2^{2k-2t-1}
\end{aligned}$$

$k \leq t :$

$$a_{i,j_{new}}^{N'-k} = a_{i,(j+s_h/2^k) \% 2^{N'-k}}^{N'-k}$$

where,

$$\begin{aligned}
j_1 &= 2^{k-t-1}j + \lfloor s_h/2^{t+1} \rfloor \\
j_2 &= 2^{k-t-2}(2j+1) + \lfloor s_h/2^{t+1} \rfloor \\
j_3 &= 2^{k-t-1}(j+1) + \lfloor s_h/2^{t+1} \rfloor
\end{aligned} \tag{3.29}$$

$k > t + 1$  :

$$\begin{aligned}
b_{i,j_{new}}^{N'-k} &= \left( \sum_{m=2^{k-t-1}i}^{2^{k-t-2}(2i+1)-1} (D_{m,j_1 \% 2^{N'-t-1}}^{N'-t-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m,n \% 2^{N'-t-1}}^{N'-t-1} \right. \\
&\quad \left. + D_{m,j_2 \% 2^{N'-t-1}}^{N'-t-1} - a_{m,j_1 \% 2^{N'-t-1}}^{N'-t-1} + a_{m,j_2 \% 2^{N'-t-1}}^{N'-t-1} \right) \\
&\quad - \sum_{m=2^{k-t-2}(2i+1)}^{2^{k-t-1}(i+1)-1} (D_{m,j_1 \% 2^{N'-t-1}}^{N'-t-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m,n \% 2^{N'-t-1}}^{N'-t-1} \\
&\quad \left. + D_{m,j_2 \% 2^{N'-t-1}}^{N'-t-1} - a_{m,j_1 \% 2^{N'-t-1}}^{N'-t-1} + a_{m,j_2 \% 2^{N'-t-1}}^{N'-t-1} \right) / 2^{2k-2t-1}
\end{aligned}$$

$k = t + 1$  :

$$\begin{aligned}
b_{i,j_{new}}^{N'-k} &= (b_{i,(j+\lfloor s_h/2^k \rfloor) \% 2^{N'-k}}^{N'-k} - c_{i,(j+\lfloor s_h/2^k \rfloor) \% 2^{N'-k}}^{N'-k} \\
&\quad + b_{i,(j+\lfloor s_h/2^k \rfloor+1) \% 2^{N'-k}}^{N'-k} + c_{i,(j+\lfloor s_h/2^k \rfloor+1) \% 2^{N'-k}}^{N'-k}) / 2
\end{aligned}$$

$k \leq t$  :

$$b_{i,j_{new}}^{N'-k} = b_{i,(j+s_h/2^k) \% 2^{N'-k}}^{N'-k}$$

where,

$$\begin{aligned}
j_1 &= 2^{k-t-1}j + \lfloor s_h/2^{t+1} \rfloor \\
j_2 &= 2^{k-t-1}(j+1) + \lfloor s_h/2^{t+1} \rfloor
\end{aligned} \tag{3.30}$$

$k > t + 1 :$

$$\begin{aligned}
c_{i,j_{new}}^{N-k} &= \left( \sum_{n=2^{k-t-1}j}^{2^{k-t-2}(2j+1)-1} (D_{i_1 \% 2^{N'-t-1},n}^{N'-t-1} + 2 \sum_{m=i_1+1}^{i_2-1} D_{m \% 2^{N'-t-1},n}^{N'-t-1} - 2 \sum_{m=i_2+1}^{i_3-1} D_{m \% 2^{N'-t-1},n}^{N'-t-1} \right. \\
&\quad \left. - D_{i_3 \% 2^{N'-t-1},n}^{N'-t-1} - b_{i_1 \% 2^{N'-t-1},n}^{N'-t-1} + 2b_{i_2 \% 2^{N'-t-1},n}^{N'-t-1} - b_{i_3 \% 2^{N'-t-1},n}^{N'-t-1} \right) \\
&\quad - \sum_{n=2^{k-t-2}(2j+1)}^{2^{k-t-1}(j+1)-1} (D_{i_1 \% 2^{N'-t-1},n}^{N'-t-1} + 2 \sum_{m=i_1+1}^{i_2-1} D_{m \% 2^{N'-t-1},n}^{N'-t-1} - 2 \sum_{m=i_2+1}^{i_3-1} D_{m \% 2^{N'-t-1},n}^{N'-t-1} \\
&\quad - D_{i_3 \% 2^{N'-t-1},n}^{N'-t-1} - b_{i_1 \% 2^{N'-t-1},n}^{N'-t-1} + 2b_{i_2 \% 2^{N'-t-1},n}^{N'-t-1} - b_{i_3 \% 2^{N'-t-1},n}^{N'-t-1})) / 2^{2k-2t-1}
\end{aligned}$$

$k = t + 1 :$

$$\begin{aligned}
c_{i,j_{new}}^{N'-k} &= (a_{(i+\lfloor s_v/2^k \rfloor) \% 2^{N'-k},j}^{N'-k} - c_{(i+\lfloor s_v/2^k \rfloor) \% 2^{N'-k},j}^{N'-k} \\
&\quad - a_{(i+\lfloor s_v/2^k \rfloor + 1) \% 2^{N'-k},j}^{N'-k} - c_{(i+\lfloor s_v/2^k \rfloor + 1) \% 2^{N'-k},j}^{N'-k}) / 2
\end{aligned}$$

$k \leq t :$

$$c_{i,j_{new}}^{N'-k} = c_{(i+s_v/2^k) \% 2^{N'-k},j}^{N'-k}$$

where,

$$\begin{aligned}
i_1 &= 2^{k-t-1}i + \lfloor s_v/2^{t+1} \rfloor \\
i_2 &= 2^{k-t-2}(2i+1) + \lfloor s_v/2^{t+1} \rfloor \\
i_3 &= 2^{k-t-1}(i+1) + \lfloor s_v/2^{t+1} \rfloor
\end{aligned} \tag{3.31}$$

On the other hand, we can verify that  $D_{i,j}^{N+h_0} = D_{\lfloor i/2^{h_0} \rfloor, \lfloor j/2^{h_0} \rfloor}^N$ , where  $0 \leq h_0 \leq h$ . This combined with (3.4) allows us to modify (3.32), (3.33) and (3.34) in such a way that avoids having to actually up-sample the signal for non-integer shifts, saving thus memory space in actual implementation, especially that the size increases exponentially. However, We have to split the equation into two

cases. The first is when  $h \geq t + 1$ , which is when the coefficients at the added levels are being used to compute  $a_{i,j_{new}}^{N'-k}$ ,  $b_{i,j_{new}}^{N'-k}$  and  $c_{i,j_{new}}^{N'-k}$ . The second is when  $t$  is large enough for the coefficients at the original levels of the tree to be used. This leads to the phase shifting relation for non-integer values as follows:

$$h \geq t + 1 :$$

$$\begin{aligned} a_{i,j_{new}}^{N'-k} &= \sum_{m=2^{k-t-1}i}^{2^{k-t-1}(i+1)-1} (D_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{j_1 \% 2^{N'-t-1}}{N'-t-1} \rfloor}^N + 2 \sum_{n=j_1+1}^{j_2-1} D_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{n \% 2^{N'-t-1}}{N'-t-1} \rfloor}^N \\ &\quad - 2 \sum_{n=j_2+1}^{j_3-1} D_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{n \% 2^{N'-t-1}}{N'-t-1} \rfloor}^N - D_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{j_3 \% 2^{N'-t-1}}{N'-t-1} \rfloor}^N) / 2^{2k-2t-1} \end{aligned}$$

$$h < t + 1 :$$

$$k > t :$$

$$\begin{aligned} a_{i,j_{new}}^{N'-k} &= \sum_{m=2^{k-t-1}i}^{2^{k-t-1}(i+1)-1} (D_{m, j_1 \% 2^{N'-t-1}}^{N'-t-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m, n \% 2^{N'-t-1}}^{N'-t-1} \\ &\quad - 2 \sum_{n=j_2+1}^{j_3-1} D_{m, n \% 2^{N'-t-1}}^{N'-t-1} - D_{m, j_3 \% 2^{N'-t-1}}^{N'-t-1} \\ &\quad - a_{m, j_1 \% 2^{N'-t-1}}^{N'-t-1} + 2a_{m, j_2 \% 2^{N'-t-1}}^{N'-t-1} - a_{m, j_3 \% 2^{N'-t-1}}^{N'-t-1}) / 2^{2k-2t-1} \end{aligned}$$

$$k \leq t :$$

$$a_{i,j_{new}}^{N'-k} = a_{i, (j+s_h/2^k) \% 2^{N'-k}}^{N'-k}$$

where,

$$\begin{aligned} j_1 &= 2^{k-t-1}j + \lfloor s_h/2^{t+1} \rfloor \\ j_2 &= 2^{k-t-2}(2j+1) + \lfloor s_h/2^{t+1} \rfloor \\ j_3 &= 2^{k-t-1}(j+1) + \lfloor s_h/2^{t+1} \rfloor \end{aligned} \tag{3.32}$$



$h \geq t+1, \quad k > t+1 :$

$$\begin{aligned}
b_{i,j_{new}}^{N'-k} = & \left( \sum_{m=2^{k-t-1}i}^{2^{k-t-2}(2i+1)-1} (D^N_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{j_1 \% 2^{N'-t-1}}{N'-t-1} \rfloor} \right. \\
& + 2 \sum_{n=j_1+1}^{j_2-1} D^N_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{n \% 2^{N'-t-1}}{N'-t-1} \rfloor} + D^N_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{j_2 \% 2^{N'-t-1}}{N'-t-1} \rfloor} \Big) \\
& - \sum_{m=2^{k-t-2}(2i+1)}^{2^{k-t-1}(i+1)-1} (D^N_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{j_1 \% 2^{N'-t-1}}{N'-t-1} \rfloor} + 2 \sum_{n=j_1+1}^{j_2-1} D^N_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{n \% 2^{N'-t-1}}{N'-t-1} \rfloor} \\
& + D^N_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{j_2 \% 2^{N'-t-1}}{N'-t-1} \rfloor} \Big) \Big) / 2^{2k-2t-1}
\end{aligned}$$

$h < t+1, \quad k > t+1 :$

$$\begin{aligned}
b_{i,j_{new}}^{N'-k} = & \left( \sum_{m=2^{k-t-1}i}^{2^{k-t-2}(2i+1)-1} (D_{m,j_1 \% 2^{N'-t-1}}^{N'-t-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m,n \% 2^{N'-t-1}}^{N'-t-1} \right. \\
& + D_{m,j_2 \% 2^{N'-t-1}}^{N'-t-1} - a_{m,j_1 \% 2^{N'-t-1}}^{N'-t-1} + a_{m,j_2 \% 2^{N'-t-1}}^{N'-t-1} \Big) \\
& - \sum_{m=2^{k-t-2}(2i+1)}^{2^{k-t-1}(i+1)-1} (D_{m,j_1 \% 2^{N'-t-1}}^{N'-t-1} + 2 \sum_{n=j_1+1}^{j_2-1} D_{m,n \% 2^{N'-t-1}}^{N'-t-1} \\
& + D_{m,j_2 \% 2^{N'-t-1}}^{N'-t-1} - a_{m,j_1 \% 2^{N'-t-1}}^{N'-t-1} + a_{m,j_2 \% 2^{N'-t-1}}^{N'-t-1} \Big) \Big) / 2^{2k-2t-1}
\end{aligned}$$

$h < t+1, \quad k = t+1 :$

$$\begin{aligned}
b_{i,j_{new}}^{N'-k} = & (b_{i,(j+\lfloor s_h/2^k \rfloor) \% 2^{N'-k}}^{N'-k} - c_{i,(j+\lfloor s_h/2^k \rfloor) \% 2^{N'-k}}^{N'-k} \\
& + b_{i,(j+\lfloor s_h/2^k \rfloor+1) \% 2^{N'-k}}^{N'-k} + c_{i,(j+\lfloor s_h/2^k \rfloor+1) \% 2^{N'-k}}^{N'-k}) / 2
\end{aligned}$$

$h < t+1, \quad k \leq t :$

$$b_{i,j_{new}}^{N'-k} = b_{i,(j+s_h/2^k) \% 2^{N'-k}}^{N'-k}$$

where,

$$\begin{aligned}
j_1 &= 2^{k-t-1}j + \lfloor s/2_h^{t+1} \rfloor \\
j_2 &= 2^{k-t-1}(j+1) + \lfloor s_h/2^{t+1} \rfloor
\end{aligned} \tag{3.33}$$

$h \geq t+1, \quad k > t+1 :$

$$\begin{aligned}
c_{i,j_{new}}^{N'-k} &= \left( \sum_{m=2^{k-t-1}i}^{2^{k-t-2}(2i+1)-1} (D^N_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{j_1 \% 2^{N'-t-1}}{N'-t-1} \rfloor} \right. \\
&+ 2 \sum_{n=j_1+1}^{j_2-1} D^N_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{n \% 2^{N'-t-1}}{N'-t-1} \rfloor} - 2 \sum_{n=j_2+1}^{j_3-1} D^N_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{n \% 2^{N'-t-1}}{N'-t-1} \rfloor} \\
&- D^N_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{j_3 \% 2^{N'-t-1}}{N'-t-1} \rfloor} \left. \right) - \sum_{m=2^{k-t-2}(2i+1)}^{2^{k-t-1}(i+1)-1} (D^N_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{j_1 \% 2^{N'-t-1}}{N'-t-1} \rfloor} \\
&+ 2 \sum_{n=j_1+1}^{j_2-1} D^N_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{n \% 2^{N'-t-1}}{N'-t-1} \rfloor} - 2 \sum_{n=j_2+1}^{j_3-1} D^N_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{n \% 2^{N'-t-1}}{N'-t-1} \rfloor} \\
&- D^N_{\lfloor \frac{m}{N'-t-1} \rfloor, \lfloor \frac{j_3 \% 2^{N'-t-1}}{N'-t-1} \rfloor} \left. \right) / 2^{2k-2t-1}
\end{aligned}$$

$h < t+1, \quad k = t+1 :$

$$\begin{aligned}
c_{i,j_{new}}^{N'-k} &= (b_{i,(j+\lfloor s_h/2^k \rfloor) \% 2^{N'-k}}^{N'-k} - c_{i,(j+\lfloor s_h/2^k \rfloor) \% 2^{N'-k}}^{N'-k} \\
&- b_{i,(j+\lfloor s_h/2^k \rfloor+1) \% 2^{N'-k}}^{N'-k} - c_{i,(j+\lfloor s_h/2^k \rfloor+1) \% 2^{N'-k}}^{N'-k}) / 2
\end{aligned}$$

$h < t+1, \quad k \leq t :$

$$c_{i,j_{new}}^{N'-k} = c_{i,(j+s_h/2^k) \% 2^{N'-k}}^{N'-k}$$

where,











$$\begin{aligned}
j_1 &= 2^{k-t-1}j + \lfloor s_h/2^{t+1} \rfloor \\
j_2 &= 2^{k-t-2}(2j+1) + \lfloor s_h/2^{t+1} \rfloor \\
j_3 &= 2^{k-t-1}(j+1) + \lfloor s_h/2^{t+1} \rfloor
\end{aligned} \tag{3.34}$$

### 3.7 EXPERIMENTAL RESULTS AND DISCUSSION

For integer shifting, our solution for Haar-domain phase shifting, given by (3.10), (3.13), and (3.16), is exact and does not incur any errors. For the subpixel case, a shift is approximated by modeling the process as upsampling by a factor of  $h$  followed by an integer shift. However, our solution does not require to actually upsample the image - i.e. the solution derived performs subpixel shifts directly from the original coefficients. For a factor of  $h$ , we can obtain a precision of  $\frac{1}{2h}$ , i.e. subpixel shifts are approximated by the closest value in multiples of  $\frac{1}{2h}$ .

In order to evaluate the accuracy of our Haar-domain phase-shifting method, we shifted the test images first by a large amount using our method. We then performed the same shift but using a successive set of smaller shifts, and used the accumulated error as a measure for performance. We performed this test on numerous images, some of which are shown in table 3.7. The performance was compared with different interpolation methods such as bilinear, bicubic, and spline. As shown in table 3.7, accumulated errors are on average an order of magnitude smaller in our method.

Table 3.1: Quantification and comparison of the accumulated residual error on several test images.

										
Bilinear	30.3943	14.0688	21.6153	21.2099	28.244	18.2492	26.867	6.7089	13.6063	21.4582
Bicubic	19.7466	6.9046	13.0134	12.7079	20.2513	8.6019	14.9709	4.7218	6.3287	12.1973
Spline	14.975	4.4063	8.9801	8.1769	15.0742	5.5216	10.162	3.9943	4.2309	7.788
Our Method	0.7646	0.7212	0.6236	0.7845	0.714	0.7215	0.6625	0.7961	0.743	0.7123

On the other hand, the low complexity of the derived solutions allow for fast processing directly in the transform domain. By examining (3.10) one can find that the complexity of evaluating  $a_{i,j_{new}}^{N-k}$  can be expressed by the difference of the bounds of the two inner sums in the equation multiplied by the difference of the bounds of the outer sum, that is  $O((j_3 - j_1) \times (2^{k-t-1}))$ . Substituting the values for  $j_1$  and  $j_3$ , the complexity is shown to be  $O(2^{k-t-1} \times 2^{k-t-1})$  when  $k > t$ . When  $k \leq t$  the complexity becomes  $O(1)$ . Therefore, one can determine that the worst case is when  $t = 0$ , that is when the shift is odd. In that case the complexity of computing  $a_{i_{new}}^{N-k}$  becomes  $O(2^{k-1} \times 2^{k-1})$ . Let  $L \times L = 2^N \times 2^N$  be the size of the two-dimensional signal, then the number of the wavelet coefficients is  $L^2 - 1 = 4^N - 1$ . The  $a$  wavelet coefficients are one third of the total number of coefficients, i.e.  $\frac{4^N - 1}{3}$ . At reduction level  $k = N$ , i.e. the root, the complexity of computing  $a_{0,0_{new}}^0$  is  $O(2^{N-1} \times 2^{N-1}) = O((\frac{L}{2})^2)$  with a probability of  $\frac{3}{L^2 - 1}$ . At the next reduction level  $k = N - 1$ , the complexity is  $O(2^{N-2} \times 2^{N-2}) = O(2^2 \times (\frac{L}{2})^2)$  with a probability of  $2^2 \times \frac{3}{L^2 - 1}$ . Table (3.7) shows the complexity and its probability at each reduction level  $k$ .

Table 3.2: Table of the complexity and probability at each reduction level  $k$  for evaluating the 2D wavelet coefficients  $a_{i,j_{new}}^{N-k}$ .

Reduction Level	Complexity	Prob.= $\frac{\text{Number of Coefficients at } k}{\text{Number of Coefficients}}$
$k = N$	$O((\frac{L}{2})^2)$	$\frac{3}{L^2-1}$
$k = N - 1$	$O((\frac{L}{2^2})^2)$	$\frac{(2^1)^2 \times 3}{L^2-1}$
$k = N - 2$	$O((\frac{L}{2^3})^2)$	$\frac{(2^2)^2 \times 3}{L^2-1}$
$k = N - 3$	$O((\frac{L}{2^4})^2)$	$\frac{(2^3)^2 \times 3}{L^2-1}$
:	:	:
$k = 1$	$O((\frac{L}{2^N})^2)$	$\frac{(2^{N-1})^2 \times 3}{L^2-1}$

By multiplying the complexities and the probabilities in table (3.7) and summing them up, the average performance of the worst case for evaluating  $a_{i,j_{new}}^{N-k}$  is found to be  $O(\lg(L))$ . Following the same analysis, one can find that the worst case complexities for evaluating  $b_{i,j_{new}}^{N-k}$  and  $c_{i,j_{new}}^{N-k}$  are found to be  $O(\lg(L))$  as well.

## 4 NON-LINEAR PHASE-SHIFTING



Figure 4.1: A scene rendered using our algorithm.

In the previous two chapters, we presented our work on linear phase-shifting of the Haar wavelets for the one-dimensional case, which can be easily generalized to  $N$  dimensions. We also presented our work for the linear two-dimensional non-separable case. In this chapter, we present an innovative method for non-linear phase-shifting that can be extended to the problem of en-

environment lighting in computer graphics. The non-linear phase-shifting in the two-dimensional signal is essentially a linear phase-shifting of three-dimensional functions, which are represented on the sphere. In other words, the non-linear phase-shifting in the two-dimensional signal is a rotation of the three-dimensional data it maps. In this chapter, we first provide an introduction to environment lighting and some background about recent related research. We then present a summary of our contributions. We also explain the applicability of non-linearly phase-shifting the 2D non-separable Haar transform signals to the problem. We finally provide experimental results and conclude with some remarks and discussion.

## 4.1 THE ENVIRONMENT LIGHTING PROBLEM

The demand for photorealism has been increasing ever since the first illumination models, which handled only point light sources. Many attempts have been made since then to increase the realism of a rendered scene. Phong [Pho75], Blinn [Bli98], and Cook and Torrance [CT81] generated glossy highlights from point sources by using general specular reflection functions that are concentrated near the mirror direction. Cook, Carpenter and Porter [CPC88] used distributed ray tracing to model glossy reflections for a specular reflection function and an arbitrary light distribution. Blinn and Newell were the first to include the light coming from the whole environment using environment maps to create mirror reflections [BN76]. Kajiya [Kaj86] introduced the *Rendering*

Equation, which provided a unified framework for rendering different types of materials.

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{H(\vec{N})} f_r(p, \omega_o, \omega_{in}) L_{in}(p, \omega_{in}) \cos \theta_{in} d\omega_{in} \quad (4.1)$$

where  $L_{in}$  is the function describing the light and  $f_r$  is the function describing the material reflection. We will describe the equation in more detail in the following section. The rendering equation provides a unified framework in the sense that  $L_{in}$  and  $f_r$  can be any function, and therefore, can represent any lighting scheme. For example, if  $f_r$  is a delta function the rendering equation simplifies to Blinn and Newell's mirror model. If  $L_{in}$  is a delta function instead, the case becomes that of a point light source. When  $f_r$  is constant the result is Lambert's law for diffuse reflection. For arbitrary lighting and reflection functions, the radiance at any point can be determined by solving the rendering equation.

## 4.2 THE RENDERING EQUATION

The rendering equation describes the interaction between incoming light and a surface material.

The following is the most general form of the equation:

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{H(\vec{N})} f_r(p, \omega_o, \omega_{in}) L_{in}(p, \omega_{in}) V(p, \omega_{in}) \cos \theta_{in} d\omega_{in} \quad (4.2)$$

where,

- $\omega_o$  is the outgoing direction.



- $L_e(p, \omega_o)$  is the emitted light at point  $p$  in the direction  $\omega_o$ .
- $L_o(p, \omega_o)$  is the radiance leaving the surface at a point  $p$  in the direction  $\omega_o$ .
- $f(p, \omega_o, \omega_{in})$  is the Bidirectional Reflectance Distribution Function (BRDF).
- $L_{in}(p, \omega_{in})$  is the incident radiance.
- $\theta_{in}$  is the angle between the unit normal  $\vec{N}$  at the point  $p$  and  $\omega_{in}$ .
- $V(p, \omega_{in})$  is the function that describes the visibility at point  $p$  along direction  $\omega_{in}$ .
- $H(\vec{N})$  is the hemisphere of directions around the normal  $\vec{N}$ .

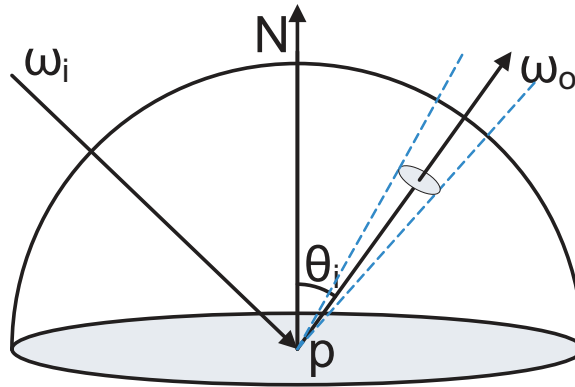


Figure 4.2: Relationship between the incident light and the light leaving a point on an object surface.

Put in words, the rendering equation integrates the product of the three contributing factors of incoming light, BRDF, and the visibility function along all incident light directions. Figure 4.2

shows the relationship between the incident light and the reflected light off the surface of an object at a given point.

## 4.3 SOLVING THE LIGHT INTEGRAL

### 4.3.1 MONTE CARLO

One of the oldest and most straightforward approaches for solving the light integral equation is to approximate the solution using the Monte Carlo method [KW86, Kaj86, DBB02]. Global illumination algorithms based on Monte Carlo are general enough to allow estimation of the integral for any material type and light distribution.

Monte Carlo is a numerical integration method that uses sampling to estimate an average solution for integration of any dimension. This is applicable to the lighting integral because the product of the light, reflection and visibility functions is too complex to evaluate using a closed-form approach.

Monte Carlo based algorithms are, however, very slow. The convergence rate for these algorithms is  $O(\frac{1}{\sqrt{n}})$ , where  $n$  is the number of samples taken to estimate the integral. This means that to cut the error in half, four times the number of samples must be taken. On the other hand, unless

sufficient light samples are taken, Monte Carlo produces noisy results that manifest as pixels that are too bright or too dark. Therefore, a substantial number of samples and accordingly more time is typically required in order to render a realistic low-noise image.

Much research has gone into improving Monte Carlo's performance without necessarily increasing the number of samples. One of the techniques that has been most effective is importance sampling [ARB03, KK03, ODJ04]. Importance sampling relies on sampling mostly in the “important” directions, which is governed by the choice of a sampling distribution function that is similar in shape to the integrand of the function that is being estimated [PH04].

Many attempts have been also made to speed the rendering time by splitting the scene synthesis into an offline prefiltering preprocess and a rendering process. Prefiltering stores the result of integrating the product of the BRDF and lighting over the visible upper hemisphere per normal direction. Cabral et al. [CON99] used prefiltering to obtain a sparse 2D set of prerendered images that were used during the rendering process to generate scenes at interactive rates. [KM00] and [KVH00] subsequently proposed alternative methods for improving prefiltering methods.

### 4.3.2 SOLVING THE INTEGRAL IN THE FREQUENCY DOMAIN

### 4.3.3 SPHERICAL HARMONICS

The first attempt to solve the integral in frequency domain was by Cabral et al. [CMS87] using *Spherical Harmonics* as basis. Spherical harmonics are the analog of the Fourier transform for representing functions on the unit sphere [Mac48]. They are the products of *Associated Legendre Functions* with functions that are periodic with respect to the azimuth angle  $\phi$ . The real spherical harmonics are defined as follows:

$$Y_l^m(\theta, \phi) = \begin{cases} \sqrt{2}K_l^m \cos(m\phi)P_l^m(\cos \theta), & m > 0 \\ \sqrt{2}K_l^m \sin(-m\phi)P_l^{-m}(\cos \theta), & m < 0 \\ K_l^0 P_l^0(\cos \theta), & m = 0 \end{cases} \quad (4.3)$$

where,

$$K_l^m = \sqrt{\frac{(2l+1)}{4\pi} \frac{(l-|m|!)}{(l+|m|!)}} \quad (4.4)$$

$l \in \mathbf{N}^+$ ,  $-l \leq m \leq l$ , and  $P_l^m(\cos \theta)$  is the associated Legendre function.

Cabral et al. [CMS87] simplified the integral by removing the emittance and the visibility functions. They also made the assumptions that the viewing direction is fixed and that the BRDF is isotropic. They then used spherical harmonics to expand the lighting function and the product of the BRDF and the cosine function, with the viewing direction as the north pole. Projecting the terms of the integral into the spherical harmonics space reduces the integration into an inner

product because of the orthonormality of spherical harmonics:

$$\begin{aligned}
L_o(p, \omega_o) &= \int_{H(\vec{N})} L_{in}(p, \omega_{in}) f_r(p, \omega_o, \omega_{in}) \cos \theta_{in} d\omega_{in} \\
&\approx \int_{H(\vec{N})} \sum_{l=0}^M \sum_{m=-l}^l a_m^l Y_m^l(\theta_{in}, \phi_{in}) \sum_{l'=0}^M \sum_{m'=-l'}^{l'} b_{m'}^{l'} Y_{m'}^{l'}(\theta_{in}, \phi_{in}) d\omega_{in} \\
&= \sum_{l=0}^M \sum_{m=-l}^l \sum_{l'=0}^M \sum_{m'=-l'}^{l'} a_m^l b_{m'}^{l'} \int_{H(\vec{N})} Y_m^l(\theta_{in}, \phi_{in}) Y_{m'}^{l'}(\theta_{in}, \phi_{in}) d\omega_{in} \\
&= \sum_{l=0}^M \sum_{m=-l}^l \sum_{l'=0}^M \sum_{m'=-l'}^{l'} a_m^l b_{m'}^{l'} \delta(l, l') \delta(m, m') \\
&= \sum_{l=0}^M \sum_{m=-l}^l a_m^l b_m^l
\end{aligned}$$

Ramamoorthi and Hanrahan [RH01] give explicit formulae in terms of the cartesian coordinates of the surface normal and approximate the solution of the integral for diffuse materials using 9 coefficients only. Their solution was the first to use spherical harmonics for photorealistic real-time rendering. Their formulae implicitly handle the required rotation of light coefficients into the local space defined by the normal as its Y-direction. This rotation is required because of the different coordinate spaces of the BRDF and the light function. The BRDF is sampled and projected into spherical harmonics' frequency domain in local space, that is, the surface normal is the Y-axis of the space. On the other hand, the light is sampled and projected in the global space, that is, the up-direction is the Y-axis of the space. Kautz et al. [KSS02] extend the use of spherical harmonics to arbitrary BRDF's under low-frequency lighting. They represent the lighting environment using 25 coefficients and rotate them for each vertex during real-time rendering at interactive rates. They also combined their method with Precomputed Radiance Transfer [SKS02] to handle interreflections and shadows. Functions represented by spherical harmonics can be ro-

tated by a linear transformation of the representation coefficients [Gre03]. The existing procedures [IR96, IR98, CJM99, KSS02], however, are slow and cause a bottleneck in the rendering process. Křivánek et al. [KKP06] propose an efficient approximation of the spherical harmonic rotation based on replacing the general spherical harmonic rotation matrix with its truncated Taylor expansion. Their proposed rotation approximation is faster and has a lower computational complexity. The approximation, however, is accurate only for small rotation angles and, therefore, applicable only to certain applications that require small successive rotation angles.

Spherical harmonics are shift-invariant, which makes them suitable for representing functions on the sphere. However, They are globally supported and, therefore, suffer from some of the same difficulties as the Fourier transform on the line such as ringing. Furthermore, spherical harmonics don't have good localization, which means a large number of coefficients is required for representing high frequency functions. Also, to handle the visibility function, it has to be combined with the BRDF and projected as one function, referred to as the *Transfer Function*, in order to be able to convert the lighting integral into an inner product. This restricts the BRDF representation in the sense that rather than having one representation for a certain material, there would be several representations of the same material, each dependant on the object occlusion properties. Furthermore, the redundancy of the BRDF information occurs per sample point per object due to the dependence of occlusion on the position of the sample point.

Imposing constraints to simplify the integral to achieve real-time or interactive rates is a necessity with spherical harmonics, when glossy or general BRDF's are used. One method is to use low

sampling rates, which is equivalent to band-limiting the illumination. This approach is used by Sloan et al. in [SKS02], [SLS03] and [SHH03]. However, band-limiting removes high frequency components, which blurs lighting detail. For diffuse materials, the error can be very low [RH01]. However, this approach is not suitable for glossy materials, which need the high frequency for an efficient representation. Another approach is to reduce the dimensionality of the problem by fixing the light or the viewing direction as in Kautz et al. [KSS02]. This approach, however, restricts the dynamics of the scene.

#### 4.3.4 HAAR WAVELETS

Ng et al. [NRH03] use non-linear wavelet approximation to achieve better localization than spherical harmonics and are successful at representing different degrees of shadowing due to wavelets' excellent capability in handling information at different scales. They, however, reduce the dimensionality of the integral to simplify it by fixing the viewing direction. Ng et al. [NRH04] develop a method for solving the triple-product integral using two-dimensional non-separable Haar wavelets to avoid reducing the dimensionality of the problem and the low sampling rate inherent in spherical harmonics methods. The following is the rendering equation reduced to a triple product:

$$L_o(p, \omega_o) = \int_S L_{in}(p, \omega_{in}) f_r(p, \omega_o, \omega_{in}) \cos \theta_{in} V(p, \omega_{in}) d\omega_{in} \quad (4.5)$$

$$= \int_S L_{in}(p, \omega_{in}) \tilde{f}_r(p, \omega_o, \omega_{in}) V(p, \omega_{in}) d\omega_{in} \quad (4.6)$$

$$\approx \int_S \left( \sum_i a_i(p) \Psi_i(\omega_{in}) \right) \left( \sum_j b_j(p, \omega_o) \Psi_j(\omega_{in}) \right) \left( \sum_k c_k(p) \Psi_k(\omega_{in}) \right) d\omega_{in} \quad (4.7)$$

$$= \sum_i \sum_j \sum_k a_i(p) b_j(p, \omega_o) c_k(p) \int_S \Psi_i(\omega_{in}) \Psi_j(\omega_{in}) \Psi_k(\omega_{in}) d\omega_{in} \quad (4.8)$$

$$= \sum_i \sum_j \sum_k C_{ijk} a_i(p) b_j(p, \omega_o) c_k(p) \quad (4.9)$$

where,

$$C_{ijk} = \int_S \Psi_i(\omega_{in}) \Psi_j(\omega_{in}) \Psi_k(\omega_{in}) d\omega_{in}$$

The above formulation is general and works for any basis. However, solving for  $C_{ijk}$  and efficiently computing the triple sum in (4.9) is not a trivial matter. One can refer to [NRH04] for an in-depth analysis of the computational complexity for the different methods to solve the triple sum.

Haar wavelets provide an efficient solution for solving the triple sum due to the following simple theorem [NRH04]:

**The Tripling Coefficient Theorem** The integral of three 2D Haar basis functions is non-zero if and only if one of the following three cases hold:

1. All three are the scaling function, that is,  $C_{ijk} = 1$ .
2. All three occupy the same wavelet square and all are different wavelet types, that is  $C_{ijk} = 2^l$ , where wavelets are at level  $l$ .



3. Two are identical wavelets, and the third is either the scaling function or a wavelet that overlaps at a strictly coarser level, that is,  $C_{ijk} = \pm 2^l$ , where the third function exists at level  $l$ .

The above theorem implies that most of the tripling coefficients are equal to zero because most pairs of basis functions don't overlap, which is the reason for the efficiency of using 2D Haar to solve the triple product.

However, Haar wavelets are not rotation-invariant, therefore, the BRDF has to be sampled per normal direction, which is storage demanding and has a limit on how much resolution it can afford. Furthermore, Haar wavelets are parametrized in cube domains to represent spherical functions. This makes it difficult or impossible to obtain an analytic rotation formula in the frequency domain. This is due to the representation mechanism, which compresses each face of the cube as a separate entity. To be able to rotate in the frequency domain data has to move from one face to another, which is not feasible with the cube representation.

Schröder and Sweldens [SS95b] construct biorthogonal wavelets on the sphere using the lifting scheme. Their method achieves better localization than spherical harmonics for high-frequency materials and is defined directly on the sphere. However, it is also not rotation-invariant and doesn't lend itself to solving the triple integral in an efficient manner.

Wang et al. [WNL06] parametrize the spherical functions using geometry maps [PH03] and provide a solution for wavelet rotation using precomputed rotation matrices. They precompute and

store the rotation matrices and use them to rotate the light coefficients into the local space of the BRDF. However, their solution is brute force rather than analytic since it would be impossible to derive rotation formulae for the frequency domain with the geometry map representation.

## 4.4 OUR CONTRIBUTIONS

**Rotation directly in the Haar wavelet domain:** We are the first to derive an explicit solution for rotating functions directly in the Haar wavelet domain. The key idea that allows us to achieve this result is the fact that for a standard non-separable Haar transform the horizontal, vertical, and diagonal coefficients are simply first order finite difference approximations of horizontal, vertical, and diagonal derivatives of the function at different scales. Therefore, we first derive the explicit expressions in the spatial domain that describe rotations of a function defined over the unit sphere. Using the chain rule, and the fact that differentiation is a linear operator, we then show how the order of rotation and differentiation can be interchanged. As a result, we derive an explicit method for rotating Haar wavelets.

**Scalable solution for light transport:** Direct Haar domain rotation completely removes the pre-computation burden and the overly expensive storage requirement, since the rotations between local and global coordinates for the BRDF, light, or visibility can be performed directly on the wavelet coefficients during run-time. Throughout this chapter the term “run-time” refers to compu-

tation of the radiance transfer without performing any precomputation. We thus call our approach a run-time radiance transfer (RRT) method as opposed to PRT. Although, our approach provides a run-time solution, as seen below, the first algorithm derived tends to be expensive. Fortunately, however, we demonstrate that a simple reformulation of the algorithm reduces the time complexity drastically. As a result, we obtain a solution that provides run-time rotation of the Haar coefficients over the entire frequency range, without requiring any data-loss, precomputation, or extensive storage: the rotated coefficients for the entire frequency range are computed recursively from only one level of coefficients of the Haar transform of the original map prior to rotation. Therefore, our solution does not have to sacrifice glossiness, against storage memory. We also don't have to interpolate between preprocessed data, whether it is the data itself or the rotation matrices, which means our method is more accurate at any given rotation angle. This is important especially with high frequency information because interpolation acts as a low-pass filter.

## 4.5 OUR METHOD

As mentioned earlier, the BRDF and the light function are represented under different coordinate spaces. The BRDF is sampled and projected into frequency domain in local space, that is, the surface normal is the Y-axis of the space. On the other hand, the light is sampled and projected in global space, that is, the up-direction is the Y-axis of the space. This requires a rotation of one of the coordinate spaces into the other before the triple product can be performed. Currently, the

most efficient computation of the triple product is performed in the Haar domain [NRH04]. Thus the key to solve the problem is to devise a method for rotating directly in the Haar domain, which we describe in the next section.

#### 4.5.1 ROTATING HAAR COEFFICIENTS

We describe our solution for the light. But, it equally applies to any function defined over the unit sphere such as the visibility map or the BRDF. Our implementation, however, rotates the BRDF data from local space into global space due to its higher degree of smoothness when mapped to a two-dimensional square, which reduces errors incurred by rotation.



Original image  $f(\theta, \phi)$

Image after rotation  $g(\theta, \phi)$

Let  $f(\theta, \phi)$  be a function that describes the light map in a spherical coordinate system at some initial orientation<sup>1</sup>. Our first goal here is to derive the relations that describe rotation in the spherical

---

<sup>1</sup>We do not explicitly include the radius in spherical coordinates to imply that we are dealing with mappings over a unit sphere.

coordinate system. Let  $\alpha_l$ ,  $\beta_l$  and  $\gamma_l$  denote the Euler rotation angles from global to local space along the cartesian axes  $X$ ,  $Y$ , and  $Z$ , respectively. Then, the light map after rotation, denoted hereafter by  $g(\theta, \phi)$ , can be derived in terms of  $f(\theta, \phi)$  as follows:

Let  $\mathbf{p}$  denote a point in the cartesian coordinate system after rotation, i.e.  $\mathbf{p} \in g(\theta, \phi)$ . Then the mapping from spherical coordinates to cartesian coordinates is given by:

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sin \theta \sin \phi \\ \cos \theta \\ \sin \theta \cos \phi \end{bmatrix}$$

where we assume a right-handed global coordinate system, with the  $Y$ -axis pointing up and the  $Z$ -axis representing the depth. The position of the point prior to an arbitrary rotation  $\mathbf{R} \in f(\theta, \phi)$  is determined by the following matrix equation:

$$\mathbf{p}' = \mathbf{R}_z \mathbf{R}_y \mathbf{R}_x \mathbf{p} = \mathbf{R} \mathbf{p} \quad (4.10)$$

where  $\mathbf{R}_x$ ,  $\mathbf{R}_y$  and  $\mathbf{R}_z$  are the familiar rotation matrices along the corresponding  $X$ ,  $Y$  and  $Z$  axes.

For brevity, we can write the elements of  $\mathbf{p}'$  as follows:

$$\mathbf{p}' = \begin{bmatrix} \mathbf{R}_1 \mathbf{p} \\ \mathbf{R}_2 \mathbf{p} \\ \mathbf{R}_3 \mathbf{p} \end{bmatrix} \quad (4.11)$$

where  $\mathbf{R}_i, i = 1, \dots, 3$  are the rows of the rotation matrix  $\mathbf{R}$ .

Assuming that the point prior to rotation has an elevation angle of  $\theta'$  and an azimuth angle of  $\phi'$ , we can readily verify that

$$g(\theta, \phi) = f(\theta', \phi')$$

where

$$\theta' = \cos^{-1}(\mathbf{R}_2 \mathbf{p}) \quad (4.12)$$

$$\phi' = \tan^{-1} \left( \frac{\mathbf{R}_1 \mathbf{p}}{\mathbf{R}_3 \mathbf{p}} \right) \quad (4.13)$$

Unfortunately, one can easily see from the above relations that the rotation is not a linear operation in the spherical coordinate system. In other words,  $\theta'$  and  $\phi'$  are not linearly related to  $\theta$  and  $\phi$ . The key observation that allows us to solve the problem is that we are specifically dealing with Haar transform coefficients, which correspond to the first order finite difference approximations of the horizontal, vertical and diagonal derivatives of the map  $f(\theta, \phi)$  at different resolution levels (scales).

Therefore, let  $\theta' = \Theta(\theta, \phi)$  and  $\phi' = \Phi(\theta, \phi)$ , denote the non-linear mappings that relate the elevation and azimuth angles before and after rotation, as given by (4.12) and (4.13). Then

$$g(\theta, \phi) = f(\Theta(\theta, \phi), \Phi(\theta, \phi)) \quad (4.14)$$

The horizontal, vertical and diagonal derivatives are essentially the derivatives with respect to the angles  $\phi$ ,  $\theta$  and  $\phi\theta$ , respectively. The illuminating feature of (4.14) is that it indicates that the

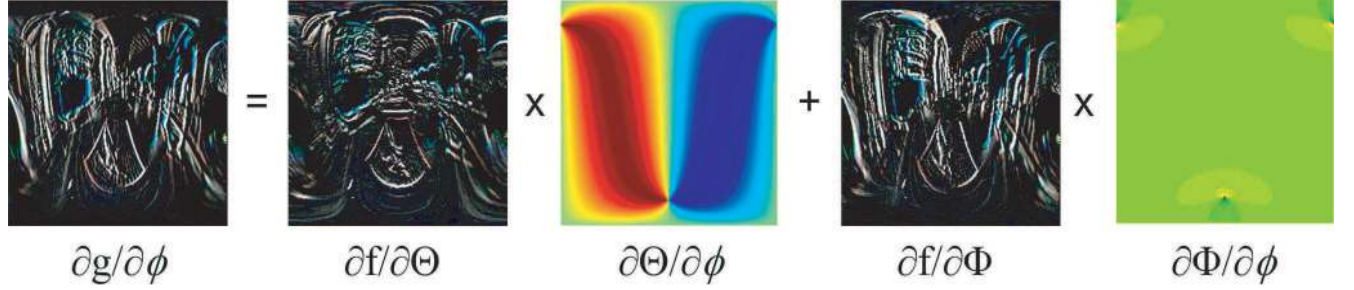


Figure 4.3: This figure shows an example of evaluating  $\frac{dg}{d\phi}$  in equation (4.16) after a rotation along the elevation angle by  $20^\circ$ .

solution to our problem lies simply in the chain rule:

$$\frac{dg}{d\theta} = \frac{\partial f}{\partial \Theta} \frac{\partial \Theta}{\partial \theta} + \frac{\partial f}{\partial \Phi} \frac{\partial \Phi}{\partial \theta} \quad (4.15)$$

$$\frac{dg}{d\phi} = \frac{\partial f}{\partial \Theta} \frac{\partial \Theta}{\partial \phi} + \frac{\partial f}{\partial \Phi} \frac{\partial \Phi}{\partial \phi} \quad (4.16)$$

$$\begin{aligned} \frac{d^2g}{d\phi d\theta} &= \frac{\partial^2 f}{\partial \Theta \partial \theta} \frac{\partial \Theta}{\partial \phi} + \frac{\partial^2 f}{\partial \Theta \partial \phi} \frac{\partial \Theta}{\partial \theta} \\ &+ \frac{\partial^2 f}{\partial \Phi \partial \theta} \frac{\partial \Phi}{\partial \phi} + \frac{\partial^2 f}{\partial \Phi \partial \phi} \frac{\partial \Phi}{\partial \theta} \end{aligned} \quad (4.17)$$

We concur from the above that finding the horizontal, vertical and diagonal Haar coefficients after some rotation reduces to the problem of applying the chain rule to the coefficients of the Haar transform of  $f$  prior to rotation - achieving, thus, rotation of the coefficients directly in the transform domain. Again, this is true because Haar coefficients conveniently correspond to function derivatives at different scales. Figure 4.3 shows an example of computing  $\frac{dg}{d\phi}$  after a vertical elevation of  $\alpha_l = 20^\circ$ , where  $\frac{\partial f}{\partial \Theta}$  and  $\frac{\partial f}{\partial \Phi}$  are obtained from the vertical coefficients  $\frac{\partial f}{\partial \theta}$  and the horizontal coefficients  $\frac{\partial f}{\partial \phi}$ , respectively, through an elevation by the same angle.

This provides an elegant analytic solution to the rotation of Haar wavelets defined over a spherical coordinate system. Furthermore, our method can be applied to any map other the longitude-latitude as long as there exists a mathematical expression that maps the data from the sphere to all pixels of a rectangular image. In other words, if  $\Theta(\theta, \phi)$  and  $\Phi(\theta, \phi)$  can be explicitly expressed mathematically then our method for rotation can be used for spherical data represented using that map.

Taking advantage of the longitude-latitude map, one can reduce the complexity of the rotation matrix  $\mathbf{R}$ . Assuming that a rotation by  $\theta_l$  along the elevation angle is aligned with the rotation around the X-axis, then  $\theta_l$  equals  $\alpha_l$ . This in turn means that a rotation by  $\theta_l$  can be represented by  $\mathbf{R}_x$ . A rotation with respect to the azimuth angle by  $\phi_l$  simply becomes a linear shift of the elevated point. This reduces  $\mathbf{R}$  from a multiplication of the three euler rotation matrices to one matrix and a linear shift. This significantly reduces the number of cosine and sine terms int  $\mathbf{R}$  and, therefore, reducing its complexity.

#### 4.5.2 MATHEMATICAL DESCRIPTION

The basic idea here is that the Haar coefficients at any level  $j + 1$  of a function  $f$  defined over the spherical coordinates can be considered as the horizontal, vertical and the diagonal derivatives of  $f$  with respect to the elevation and azimuth angles at that level (scale). Therefore, given the Haar coefficients of  $f$  at level  $j + 1$ , we can directly compute the derivatives  $g_\theta(\theta, \phi) = \frac{dg}{d\theta}$ ,  $g_\phi(\theta, \phi) =$



$\frac{dg}{d\phi}$ , and  $g_{\theta\phi}(\theta, \phi) = \frac{d^2g}{d\phi d\theta}$  of the rotated function  $g$  at level  $j+1$  by simply using the equations (4.15)-(4.17). To obtain the Haar coefficients of  $g$  at level  $j$  we simply need to convolve these derivatives with  $2 \times 2$  averaging kernels and then downsample the results.

Simple inspection then shows that Haar coefficient at all coarser levels  $j-1, j-2, j-3, \dots$  can be computed directly from the derivatives  $g_\theta$ ,  $g_\phi$ , and  $g_{\theta\phi}$  by a series of convolutions and downsampling. We describe the main idea using the vertical coefficients as an example. Suppose we want to compute the vertical coefficients  $v_{j-l}$  at level  $j-l$  using the vertical derivatives  $g_\theta$ . This amounts to convolving the vertical derivatives with two separate kernels as follows:

$$v_{j-l}(\theta, \phi) = \sum_{\rho} h_{s,j-l}(\rho - \phi) \sum_{\tau} g_{\theta}(\theta, \phi) h_{t,j-l}(\tau - \theta) \quad (4.18)$$

where

$$h_{t,j-l}(\theta) = \left(2^{l+1} - |\theta|\right) \quad , \quad \theta \in [-2^{l+1} + 1, 2^{l+1} - 1] \quad (4.19)$$

and

$$h_{s,j-l}(\phi) = \frac{1}{2^{l+2}} \quad , \quad \phi \in [-2^{l+1} + 1, 2^{l+1} - 1] \quad (4.20)$$

followed by downsampling by a factor of  $2^{l+2}$ .

Horizontal Haar coefficients at level  $j-l$  can be similarly computed from  $g_\phi$  by interchanging the role of  $h_t$  and  $h_s$ . The diagonal coefficients are simply computed by taking the cross-derivative either using the horizontal or the vertical coefficients and convolving in both directions by  $h_s$ .

### 4.5.3 ALGORITHMIC OPTIMIZATION

The simple but powerful results derived in the previous section show that we can directly compute the Haar coefficients of a rotated function at all levels  $j - l$ ,  $l = 0, 1, 2, \dots$  directly from the Haar coefficients of the original function at only one level, i.e. level  $j + 1$ . However, it can be verified from (4.18)-(4.20) that as  $l$  increases the time complexity of (4.18) increases exponentially. At a first glance, this may seem disappointing, however, it turns out that this is rather misleading, and there is a simple solution to reduce the complexity to provide run-time computation.

There are two measures that we can take to optimize the algorithm: First, note that the Haar coefficients at all levels  $j - l$ ,  $l = 0, 1, 2, \dots$  are computed from the derivatives at level  $j + 1$ . Second the convolution kernels  $h_{t,j-l}$  and  $h_{s,j-l}$  increase exponentially in size as  $l$  increases. Both problems can be alleviated by identifying the fact that the proposed computations in the previous section can be performed recursively so that at each level the computation of the Haar coefficients depends only on the Haar coefficients computed for the previous level. To this end, we note that for all  $l \geq 1$ , the kernel  $h_{t,j-l}$  can be written as the convolution of two kernels:

$$h_{t,j-l} = h_{t,j} * h_{t,j}^l \quad (4.21)$$

where

$$h_{t,j}^l(\theta) = \begin{cases} h_{t,j}\left(\frac{\theta}{2^l}\right) & \text{if } \frac{\theta}{2^l} \text{ is an integer} \\ 0 & \text{otherwise} \end{cases} \quad (4.22)$$

Note that except for three values, all the values in the kernel  $h_{t,j}^l$  are zero. Essentially,  $h_{t,j}^l$  is the same kernel as  $h_{t,j}$ , but upsampled by a factor of  $2^l$ , by zero padding all elements between the non-

zero values. Therefore, the algorithm can be implemented recursively by repeated convolution with the upsampled kernel  $h_{t,j}^l$  for all  $l \geq 1$ . The latter is very cheap since except for three values all the values in the kernel are zero. Simple inspection indicates that the reduced algorithm is  $O(n)$ .

In addition to the above algorithmic optimization for recursive computation of the Haar coefficients, we can further reduce the computational time by modifying also the rotation step based on the chain rule described in Section 4.5.1. To this end note that without loss of generality we can assume that the function to be rotated is initially aligned such that the  $x$ -axis coincide with the  $\theta$ -axis. The consequence of this assumption is that all rotations reduce to rotation around the  $\theta$ -axis (or the  $x$ -axis) followed by a simple shifting along the  $\phi$ -axis, which is extremely cheap (i.e.  $O(1)$ ).

The following algorithm summarizes our method:

- For each pixel, determine the local outgoing direction and retrieve the BRDF transformed map corresponding to that direction.
- Assuming the transformed BRDF data has  $0 \dots n - 1$  levels of resolution, the vertical, horizontal and diagonal coefficients at level  $n - 1$  are rotated by  $\phi_{\vec{N}}$  and  $\theta_{\vec{N}}$ , which are the azimuth and elevation angles of the normal  $\vec{N}$  at the current pixel.
- The coefficients at the lower resolution levels are subsequently evaluated by recursively convolving by the filters  $h_s$  and  $h_t$ , where  $h_s$  and  $h_t$  have a size of 2 and 3 coefficients respectively.

- The rotated data is plugged in the triple integral computation.

The above algorithm has a complexity of  $O(N)$ , where  $N$  is the number of coefficients. One might argue that this is the same complexity as saving multiple levels of resolution of untransformed data using a mipmap, rotating the required level spatially and then transforming. Representing the data using a wavelet transform, however, is more compact than a mipmap, which saves storage space as is our goal. It is also more convenient to store the data in its transformed state to be readily available for use for the triple product computation.

We would also like to mention that one can start at any resolution level that is lower than  $n - 1$ . In that case, the computational complexity is reduced to  $O(N/4^k)$ , where  $k$  is the number of levels removed.

## 4.6 EXPERIMENTAL RESULTS AND DISCUSSION

Our algorithm is implemented using CUDA ("Compute Unified Device Architecture") in combination with Cg for graphics rendering. The algorithm is implemented into the following passes:

**The Cg Pass:** This pass takes advantage of the interpolator and rasterizer of the graphic pipeline to output a fragment buffer that contains the interpolation information, vertex id, and object id per pixel.

**The Rotation Input Pass:** This pass outputs a buffer which contains the BRDF tile id and the rotation angles required per non-background pixel.

**The Rotation Pass:** This pass is actually divided into three passes that run concurrently to optimize speed. The three passes evaluate the horizontal, vertical and diagonal coefficients of the rotated BRDF tiles.

**The Triple Integral Pass:** This pass evaluates the illumination of each non-background pixel using the triple sum.

To evaluate the error incurred from rotation, we preprocessed the BRDF data by rotating it spatially. We also preprocessed the BRDF data by rotating it using our method. This allowed us to create images that are comparable. Figures (4.5) and (4.6) show our error with respect to spatial rotation. The figures show results from rotating the data from levels 5 and 6 and recursively evaluating the rest of the coefficients. Rotating from level 5 is slightly better because it has less accumulation error because of the smaller number of levels. However, that doesn't necessarily mean that a smaller number of levels is better. The smaller the number of coefficients at a certain level the less smooth the data becomes, which causes inaccuracy due to discontinuity.

Our most important contribution is providing the first real solution to rotating Haar wavelets. Although one method of rotation already exists, it does so by creating a rotation matrix per discretized rotation angle. Each rotation matrix is created during a preprocess stage where each wavelet of a resolution less than or equal the required image size is rotated spatially then transformed and stored

as one column of that specific rotation matrix. The rotation matrices are then used during rendering time to rotate any of the required data. This solution is computational rather than analytical, therefore, it relies on preprocessing and discretization, which produce errors due to interpolation.

We are currently able to rotate coefficients at level 4 using the GPU and 5 using the CPU. We recursively generate the coefficients at the coarser levels on the GPU. This is the equivalent of using 256 and 1024 coefficients for rendering respectively. Although PRT methods are capable of linearly compressing data and generating glossier materials at the current time, they still suffer from excess storage, discretization and interpolation error. It is a well-known fact that the speed of computing systems increase at a higher rate than memory, especially now that the market is focusing on parallel computing both on the CPU and GPU. In a few years time our method will be able to render using a larger number of coefficients and achieve interactive/real time.

## 4.7 CONCLUSION

In this thesis, we aspired to establish the grounds for our work by giving a brief but necessary introduction to wavelets, their different properties and the state of wavelet related research on the property of shift-invariance. Achieving shift-invariance was, up until now, the only method to achieve phase-shifting. However, that compromised other desired properties, which we wish to keep for the purposes of the environment lighting application at hand. These properties, namely,

orthogonality, perfect reconstruction and localization are essential to solving the triple product that results from projecting the lighting integral terms into frequency space and, therefore, are a necessity to preserve. We present our current work for phase-shifting, which does not rely on shift-invariance and therefore preserves these important properties. Our recent work on phase-shifting for one-dimensional and two-dimensional signals was demonstrated in chapters two and three. In this chapter, we discussed the environment lighting problem and its relationship to Haar wavelets. Our goal is to achieve realistic rendering of synthetic scenes using Haar wavelets as the integration medium while improving the efficiency in terms of storage over other methods by means of rotating during rendering time. In order to do so we devised a novel method to tackle the non-linear phase-shifting that makes use of the fact that the Haar transform implicitly contains the horizontal, vertical and diagonal derivatives of the signal. Our run-time radiance transfer method (RRT) provides an elegant solution to the non-trivial problem of rotating Haar wavelets, providing thus a solution for run-time computation of light transport based on the rendering equation. The resulting algorithm scales nicely so that no trade-off is required in terms of storage, computational cost, and bandwidth. Furthermore, since the proposed solution does not require precomputation, errors due to interpolation are avoided.

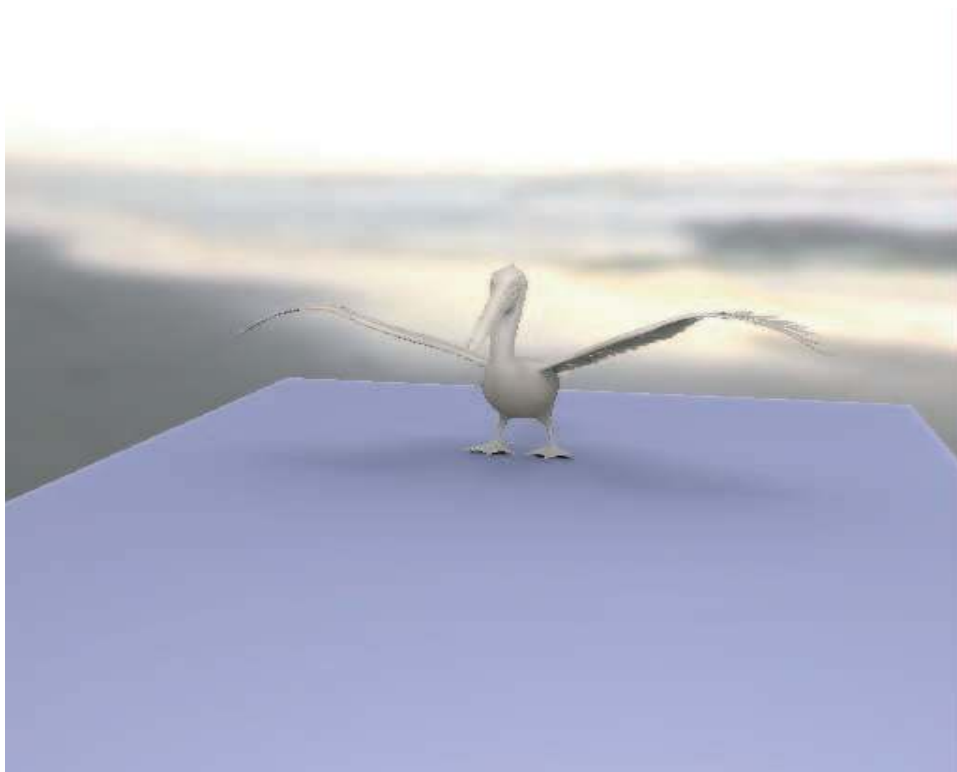


Figure 4.4: The same scene rendered using our algorithm with and without texture for better



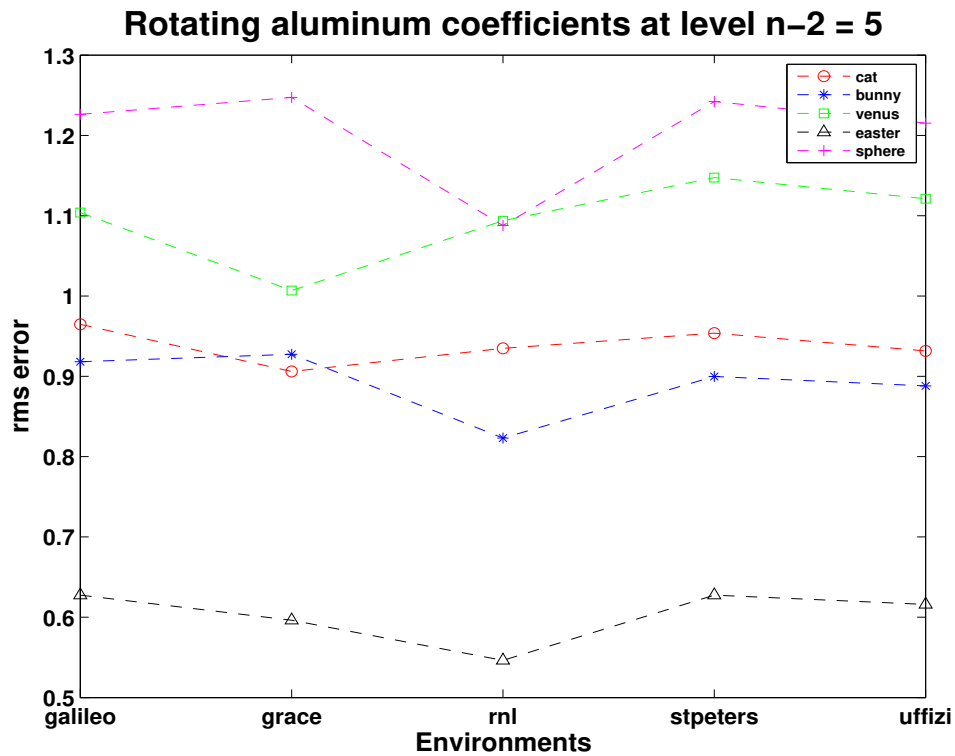
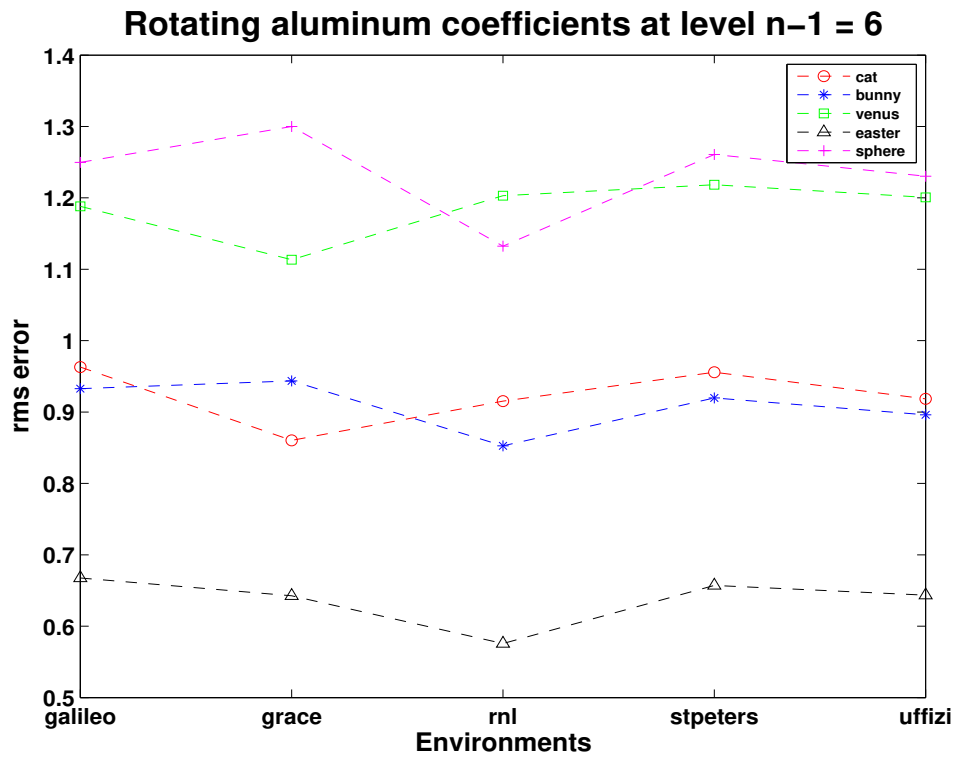


Figure 4.5: The above graphs show the rms error comparing rendered images of different models using measured a aluminum bronze material under different environments. The sets of images are generated using preprocessed data under spatial rotation versus our method of rotation. Using our method, we rotated the coefficients at level 6 then evaluated the coefficients at the coarser levels

recursively

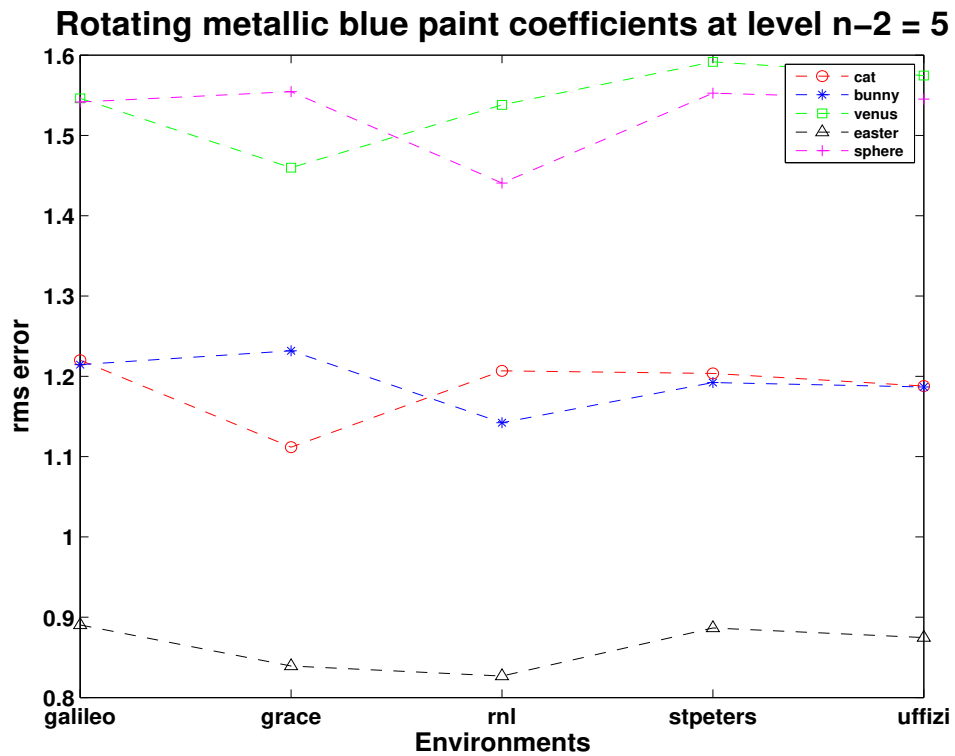
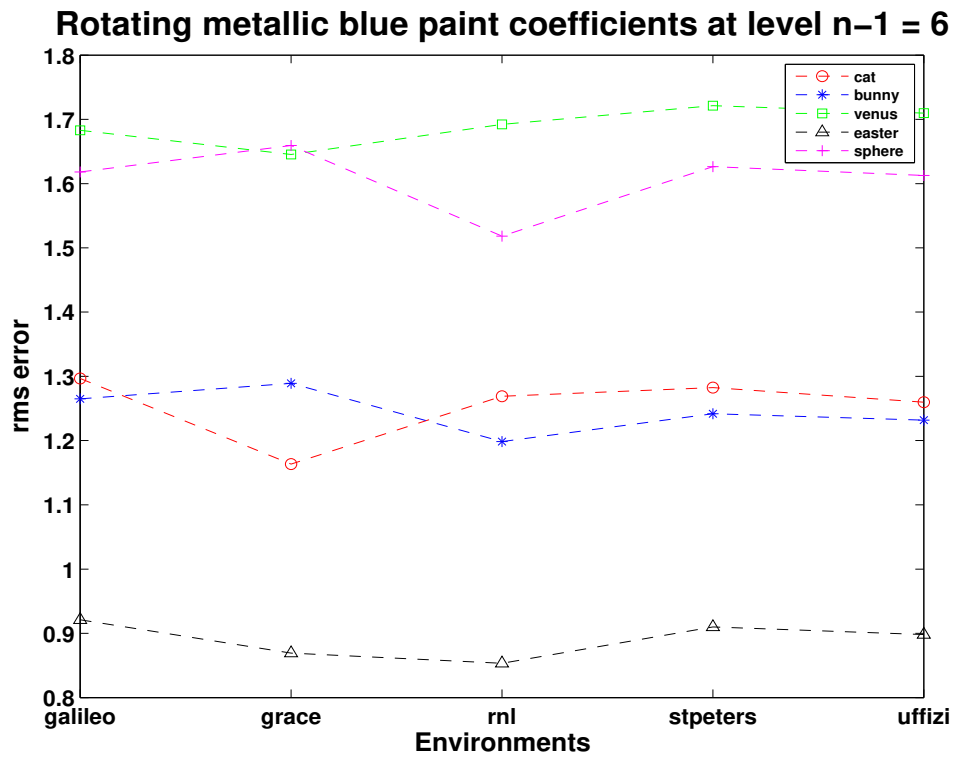


Figure 4.6: The above graphs show the rms error comparing rendered images of different models using measured a blue metallic paint material under different environments. The sets of images are generated using preprocessed data under spatial rotation versus our method of rotation. Using our method, we rotated the coefficients at level 6 then evaluated the coefficients at the coarser levels

recursively



## LIST OF REFERENCES

- [AF08a] Mais Alnasser and Hassan Foroosh. “Phase-Shifting for Non-Separable 2D Haar Wavelets.” *IEEE Transactions on Image Processing*, 2008.
- [AF08b] Mais Alnasser and Hassan Foroosh. “Phase-Shifting in the Discrete Haar Domain.” *Submitted to IEEE Transactions on Image Processing*, 2008.
- [ARB03] Sameer Agarwal, Ravi Ramamoorthi, Serge Belongie, and Henrik Wann Jensen. “Structured importance sampling of environment maps.” In *SIGGRAPH ’03: ACM SIGGRAPH 2003 Papers*, pp. 605–612, New York, NY, USA, 2003. ACM Press.
- [BC90] Michael Banks and Elaine Cohen. “Real time spline curves from interactively sketched data.” In *SI3D ’90: Proceedings of the 1990 symposium on Interactive 3D graphics*, pp. 99–107, New York, NY, USA, 1990. ACM Press.
- [BGG97] C. S. Burrus, R. A. Gopinath, and H. Guo. *Introduction to Wavelets and Wavelets Transforms*. Prentice Hall, August 1997.
- [Bli98] James F. Blinn. “Models of light reflection for computer synthesized pictures.” pp. 103–109, 1998.
- [BN76] James F. Blinn and Martin E. Newell. “Texture and reflection in computer generated images.” *Commun. ACM*, **19**(10):542–547, 1976.
- [CD95] R. R. Coifman and D. L. Donoho. “Translation-Invariant De-Noising.” Technical report, Department of Statistics, 1995.
- [CG91] George Celniker and Dave Gossard. “Deformable curve and surface finite-elements for free-form shape design.” In *SIGGRAPH ’91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pp. 257–266, New York, NY, USA, 1991. ACM Press.

- [Chu92] C. K. Chui. *An introduction to wavelets*. Academic Press Professional, Inc., San Diego, CA, USA, 1992.
- [CJM99] Choi C., Ivanic J., Gordon M., and Ruedenberg K. “Rapid and stable determination of rotation matrices between spherical harmonics by direct recursion.” *The Journal of Chemical Physics* 111, **19**:8825–8831, November 1999.
- [CK93] T. Chang and C. C. J. Kuo. “Texture analysis and classification with tree-structured wavelet transform.” *IEEE Transactions on Image Processing*, **2**(4):429–441, 1993.
- [CMS87] Brian Cabral, Nelson Max, and Rebecca Springmeyer. “Bidirectional reflection functions from surface bump maps.” *SIGGRAPH Comput. Graph.*, **21**(4):273–281, 1987.
- [Coh92] Michael F. Cohen. “Interactive spacetime control for animation.” In *SIGGRAPH ’92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pp. 293–302, New York, NY, USA, 1992. ACM Press.
- [CON99] Brian Cabral, Marc Olano, and Philip Nemec. “Reflection space image based rendering.” In *SIGGRAPH ’99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 165–170, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [CPC88] R. L. Cook, T. Porter, and L. Carpenter. “Distributed ray tracing.” pp. 139–147, 1988.
- [CSS96] Per H. Christensen, Eric J. Stollnitz, and David H. Salesin. “Global Illumination of Glossy Environments Using Wavelets and Importance.” *ACM Transactions on Graphics*, **15**(1):37–71, 1996.
- [CT81] Robert L. Cook and Kenneth E. Torrance. “A reflectance model for computer graphics.” In *SIGGRAPH ’81: Proceedings of the 8th annual conference on Computer graphics and interactive techniques*, pp. 307–316, New York, NY, USA, 1981. ACM Press.
- [CW93] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Boston, MA, 1993.
- [CW94] R. R. Coifman and M. V. Wickerhauser. “Adapted Waveform Analysis as a Tool for Modeling, Feature Extraction, and Denoising.” *Optical Engineering*, **33**(7):2170–2174, 1994.
- [Dau92] Ingrid Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

- [DAV99] Minh Do, Serge Ayer, and Martin Vetterli. “Invariant Image Retrieval Using Wavelet Maxima Moment.” In *Visual Information and Information Systems*, pp. 451–458, 1999.
- [DBB02] Philip Dutre, Kavita Bala, and Philippe Bekaert. *Advanced Global Illumination*. A. K. Peters, Ltd., Natick, MA, USA, 2002.
- [DH92] P.-E. Danielsson and M. Hammerin. “High-accuracy rotation of images.” *CVGIP: Graph. Models Image Process.*, **54**(4):340–344, 1992.
- [DJL92] Ronald A. DeVore, Björn Jawerth, and Bradley J. Lucier. “Surface compression.” *Comput. Aided Geom. Des.*, **9**(3):219–239, 1992.
- [Don95] D. L. Donoho. “De-noising by Soft-Thresholding.” *IEEE Transactions on Information Theory*, **41**(3):613–627, 1995.
- [Dut89] P. Dutilleux. “An Implementation of the ”algorithme à trous” to Compute the Wavelet Transform.” In J.-M. Combes, A. Grossmann, and P. Tchamitchian, editors, *Wavelets. Time-Frequency Methods and Phase Space*, pp. 298–+, 1989.
- [DV] M. Do and M. Vetterli. “Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden Markov models.”
- [DV00] M. Do and M. Vetterli. “Texture similarity measurement using Kullback-Leibler distance on wavelet subbands.”, 2000.
- [EDD95] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. “Multiresolution analysis of arbitrary meshes.” In *SIGGRAPH ’95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 173–182, New York, NY, USA, 1995. ACM Press.
- [ES95] T. D. DeRose E. J. Stollnitz and D. H. Salesin. “Wavelets for Computer Graphics: A Primer, Part 1.” *IEEE Computer Graphics and Applications*, **15**(3):76–84, 1995.
- [FB88] David R. Forsey and Richard H. Bartels. “Hierarchical B-spline refinement.” In *SIGGRAPH ’88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pp. 205–212, New York, NY, USA, 1988. ACM Press.
- [FBN07] M. Figueiredo, J. Bioucas-Dias, and R. Nowak. “Majorization-Minimization Algorithms for Wavelet-Based Image Restoration.” *Preprint*, 2007.
- [FS94] Adam Finkelstein and David H. Salesin. “Multiresolution Curves.” *Computer Graphics*, **28**(Annual Conference Series):261–268, 1994.



- [Gab46] Dennis Gabor. “Theory of Communication.” *Journal of the Institute of Electrical Engineers*, **93**(22):429–457, 1946.
- [GB92] R. A. Gopinath and C. S. Burrus. “Wavelets and Filter Banks.” In C. K. Chui, editor, *Wavelets: A Tutorial in Theory and Applications*, pp. 603–654. Academic Press, San Diego, CA, 1992.
- [GGM84] P. Goupillaud, A. Grossmann, and J. Morlet. “Cycle-Octave and related transforms in seismic signal analysis.” *Geoexploration*, **23**:85–102, 1984.
- [GM84] A. Grossmann and J. Morlet. “Decomposition of Hardy functions into square integrable wavelets of constant shape.” **15**:723–736, 1984.
- [Gre03] Robin Green. “Spherical Harmonic Lighting: The Gritty Details.” Technical report, Sony Computer Entertainment America, 2003.
- [GSC93] Steven J. Gortler, Peter Schröder, Michael F. Cohen, and Pat Hanrahan. “Wavelet radiosity.” In *SIGGRAPH ’93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pp. 221–230, New York, NY, USA, 1993. ACM Press.
- [Haa10] A. Haar. “Zur Theorie der orthogonalen Funktionensysteme.” *Mathematische Annalen*, pp. 331–371, 1910.
- [HDD94] Hugues Hoppe, Tony DeRose, Tom Duchamp, Mark Halstead, Hubert Jin, John McDonald, Jean Schweitzer, and Werner Stuetzle. “Piecewise Smooth Surface Reconstruction.” *Computer Graphics*, **28**(Annual Conference Series):295–302, 1994.
- [HN05] J. Haupt and R. Nowak. “Signal Reconstruction from Noisy Random Projections.” *IEEE Trans. on Information Theory*, 2005.
- [HSA91] Pat Hanrahan, David Salzman, and Larry Aupperle. “A rapid hierarchical radiosity algorithm.” In *SIGGRAPH ’91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pp. 197–206, New York, NY, USA, 1991. ACM Press.
- [HW95] D. Donoho H. Krim, S. Mallat and A.S. Willsky. “Best basis algorithm for signal enhancement.” *icassp*, **3**:1561–1564, 1995.
- [IR96] Joseph Ivanic and Klaus Ruedenberg. “Rotation matrices for real spherical harmonics. Direct determination by recursion.” *J. Phys. Chem. A*, **100**(15):6342–6347, 1996.

- [IR98] Joseph Ivanic and Klaus Ruedenberg. “Correction: Rotation matrices for real spherical harmonics. Direct determination by recursion.” *J. Phys. Chem. A*, **102**(45):9099–9100, 1998.
- [Kaj86] James T. Kajiya. “The rendering equation.” In *SIGGRAPH ’86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pp. 143–150, New York, NY, USA, 1986. ACM Press.
- [KG85] H. Kieseletter and A. Graf. “Rotation of digital grids and corresponding models.” Zentral Institut fur Kybernetik und Informations Prozesse, Akademie der Wissenschaften der DDR Tech. Rep., 1985.
- [KK03] Thomas Kollig and Alexander Keller. “Efficient illumination by high dynamic range images.” In *EGRW ’03: Proceedings of the 14th Eurographics workshop on Rendering*, pp. 45–50, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [KKP06] Jaroslav Křivánek, Jaakko Konttinen, Sumanta Pattanaik, Kadi Bouatouch, and Jiří Žára. “Fast approximation to spherical harmonics rotation.” In *SIGGRAPH ’06: ACM SIGGRAPH 2006 Sketches*, p. 154, New York, NY, USA, 2006. ACM Press.
- [KM00] Jan Kautz and Michael D. McCool. “Approximation of Glossy Reflection with Prefiltered Environment Maps.” In *Graphics Interface*, pp. 119–126, 2000.
- [KSS02] Jan Kautz, Peter-Pike Sloan, and John Snyder. “Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics.” In *EGRW ’02: Proceedings of the 13th Eurographics workshop on Rendering*, pp. 291–296, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.
- [KVH00] Jan Kautz, Pere-Pau Vázquez, Wolfgang Heidrich, and Hans-Peter Seidel. “Unified Approach to Prefiltered Environment Maps.” In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, pp. 185–196, London, UK, 2000. Springer-Verlag.
- [KW86] Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo methods. Vol. 1: basics*. Wiley-Interscience, New York, NY, USA, 1986.
- [LGO96] M. Lang, H. Guo, J. E. Odegard, C. S. Burrus, and R. O. Wells. “Noise Reduction Using an Undecimated Discrete Wavelet Transform.” *IEEE Signal Processing Letters*, **3**(1), 1996.
- [Mac48] T. M. MacRobert. *Spherical Harmonics; an elementary treatise on harmonic functions, with applications*. Dover Publications, 1948.



- [Mal89] S. G. Mallat. “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation.” *IEEE Trans. Pattern Anal. Mach. Intell.*, **11**(7):674–693, 1989.
- [Mal91] Stephane G. Mallat. “Zero-Crossings of a Wavelet Transform.” **37**(4):1019–1033, July 1991.
- [Mal98] S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1998.
- [MK96] J. Magarey and N. Kingsbury. “Motion estimation using complex wavelets.” In *ICASSP '96: Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference Proceedings., 1996 IEEE International Conference*, pp. 2371–2374, Washington, DC, USA, 1996. IEEE Computer Society.
- [MM96] B. S. Manjunath and W. Y. Ma. “Texture Features for Browsing and Retrieval of Image Data.” *IEEE Trans. Pattern Anal. Mach. Intell.*, **18**(8):837–842, 1996.
- [MSS92] David Meyers, Shelley Skinner, and Kenneth Sloan. “Surfaces from contours.” *ACM Trans. Graph.*, **11**(3):228–258, 1992.
- [MT89] J. Morlet M. Holschneider, R. Kronland-Martinet and P. Tchamitchian. “A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform.” In J.-M. Combes, A. Grossmann, and P. Tchamitchian, editors, *Wavelets. Time-Frequency Methods and Phase Space*, pp. 286–+, 1989.
- [MW07] R. Nowak M. Figueiredo and S. Wright. “Gradient Projection for Sparse Reconstruction: Applications to Compressed Sensing and Other Inverse Problems.” *Preprint*, 2007.
- [New93] D. E. Newland. “Harmonic Wavelet Analysis.” In *Proc. R. Soc. Lond. A*, volume 443, pp. 203–225, 1993.
- [New99] D. E. Newland. “Harmonic wavelets in vibrations and acoustics.” *Phil. Trans. R. Soc. Lond. A*, **357**(1760):2607–2625, 1999.
- [NRH03] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. “All-frequency shadows using non-linear wavelet lighting approximation.” In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pp. 376–381, New York, NY, USA, 2003. ACM Press.
- [NRH04] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. “Triple product wavelet integrals for all-frequency relighting.” In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pp. 477–487, New York, NY, USA, 2004. ACM Press.

- [ODJ04] Victor Ostromoukhov, Charles Donohue, and Pierre-Marc Jodoin. “Fast hierarchical importance sampling with blue noise properties.” In *SIGGRAPH ’04: ACM SIGGRAPH 2004 Papers*, pp. 488–495, New York, NY, USA, 2004. ACM Press.
- [Pae86] A. W. Paeth. “A fast algorithm for general raster rotation.” In *Proceedings on Graphics Interface ’86/Vision Interface ’86*, pp. 77–81, Toronto, Ont., Canada, Canada, 1986. Canadian Information Processing Society.
- [PH03] Emil Praun and Hugues Hoppe. “Spherical parametrization and remeshing.” In *SIGGRAPH ’03: ACM SIGGRAPH 2003 Papers*, pp. 340–349, New York, NY, USA, 2003. ACM Press.
- [PH04] Matt Pharr and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [Pho75] Bui Tuong Phong. “Illumination for Computer Generated Pictures.” *Commun. ACM*, **18**(6):311–317, 1975.
- [PS83] Michael Plass and Maureen Stone. “Curve-fitting with piecewise parametric cubics.” *SIGGRAPH Comput. Graph.*, **17**(3):229–239, 1983.
- [RH01] Ravi Ramamoorthi and Pat Hanrahan. “An efficient representation for irradiance environment maps.” In *SIGGRAPH ’01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 497–500, New York, NY, USA, 2001. ACM Press.
- [RK99] P. de Rivaz and N. Kingsbury. “Complex Wavelet Features for Fast Texture Image Retrieval.” In *Proc. ICIP*, pp. I:109–113, 1999.
- [RK00] P. De Rivaz and N. G. Kingsbury. “Fast Segmentation Using Level Set Curves of Complex Wavelet Surfaces.” In *ICIP*, 2000.
- [RN06] J. Jaupt R. Castro and R. Nowak. “Compressed Sensing vs. Active Learning.” In *Proc. IEEE International Conf. on Accoustics, Speech, and Signal Processing (ICASSP)*, 2006.
- [SB97] H. Sari-Sarraf and D. Brzakovic. “A Shift-Invariant Discrete Wavelet Transform.” *IEEE Transactions on Signal Processing*, **45**(10):2621–2626, 1997.
- [SC94] N. Saito and R. R. Coifman. “Local discriminant bases.” In A. F. Laine and M. A. Unser, editors, *Wavelet Applications in Signal and Image Processing II, Proc. SPIE 2303*, pp. 2–14, 1994.

- [SC95] N. Saito and R.R. Coifman. “On local orthonormal bases for classification and regression.” *icassp*, **3**:1529–1532, 1995.
- [SF95] E. P. Simoncelli and W. T. Freeman. “The Steerable Pyramid: A Flexible Architecture for Multi-Scale Derivative Computation.” In *International Conference on Image Processing*, volume 3, pp. 444–447, 23-26 Oct. 1995, Washington, DC, USA, 1995.
- [SFA92] Eero P. Simoncelli, William T. Freeman, Edward H. Adelson, and David J. Heeger. “Shiftable Multi-scale Transforms.” *IEEE transactions on informations theory*, **38**(2), 1992.
- [SGC93] Peter Schröder, Steven J. Gortler, Michael F. Cohen, and Pat Hanrahan. “Wavelet Projections for Radiosity.” In *Fourth Eurographics Workshop on Rendering*, number Series EG 93 RW, pp. 105–114, Paris, France, 1993.
- [SGC94] Peter Schröder, Steven J. Gortler, Michael F. Cohen, and Pat Hanrahan. “Wavelet Projections for Radiosity.” *Computer Graphics Forum*, **13**(2):141–151, 1994.
- [SHH03] Peter-Pike Sloan, Jesse Hall, John Hart, and John Snyder. “Clustered principal components for precomputed radiance transfer.” In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pp. 382–391, New York, NY, USA, 2003. ACM Press.
- [SKS02] Peter-Pike Sloan, Jan Kautz, and John Snyder. “Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments.” In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 527–536, New York, NY, USA, 2002. ACM Press.
- [SLS03] Peter-Pike Sloan, Xinguo Liu, Heung-Yeung Shum, and John Snyder. “Bi-scale radiance transfer.” *ACM Trans. Graph.*, **22**(3):370–375, 2003.
- [SN97] Gilbert Strang and Truong Nguyen. *Wavelets and Filter Banks*. Wellesley, 1997. StRA g 97:1 1.Ex.
- [SS95a] Peter Schröder and Wim Sweldens. “Spherical wavelets: efficiently representing functions on the sphere.” *Computer Graphics*, **29**(Annual Conference Series):161–172, 1995.
- [SS95b] Peter Schröder and Wim Sweldens. “Spherical wavelets: efficiently representing functions on the sphere.” In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 161–172, New York, NY, USA, 1995. ACM Press.

- [Str89] G. Strang. “Wavelets and dilation equations: a brief introduction.” *SIAM Rev.*, **31**(4):614–627, 1989.
- [UTY95] M. Unser, P. Thévenaz, and L. P. Yaroslavsky. “Convolution-based interpolation for fast, high-quality rotation of images.” *IEEE Transactions on Image Processing*, **4**(10):1371–1381, 1995.
- [Vai93] P. P. Vaidyanathan. *Multirate systems and filter banks*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.
- [Vet86] M Vetterli. “Filter banks allowing perfect reconstruction.” *Signal Process.*, **10**(3):219–244, 1986.
- [Vet87] M. Vetterli. “A Theory of Multirate Filter Banks.” **35**(3):356–372, Mar. 1987.
- [VK95] Martin Vetterli and Jelena Kovačević. *Wavelets and subband coding*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.
- [Wei95] Karl Weierstrass. *Mathematische Werke, volume II*. Mayer & Muller, Berlin, 1895.
- [WNL06] R. Wang, R. Ng, D. Luebke, and G. Humphreys. “Efficient wavelet rotation for environment map rendering.” In *Proceedings of the Eurographics Symposium on Rendering*, 2006.
- [WW92] William Welch and Andrew Witkin. “Variational surface modeling.” In *Computer Graphics (Proc. SIGGRAPH '92) Graphics*, volume 26, 1992.