

2006

## Some new methods for image compression

Muhammad Aslam  
*West Virginia University*

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

---

### Recommended Citation

Aslam, Muhammad, "Some new methods for image compression" (2006). *Graduate Theses, Dissertations, and Problem Reports*. 4212.  
<https://researchrepository.wvu.edu/etd/4212>

This Dissertation is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Dissertation has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact [researchrepository@mail.wvu.edu](mailto:researchrepository@mail.wvu.edu).

Some New Methods for Image Compression

Muhammad Aslam

Dissertation submitted to the  
Eberly College of Arts and Sciences  
at West Virginia University  
in partial fulfillment of the requirements  
for the degree of

Doctor of Philosophy  
in  
Mathematics

Sherman D. Riemenschneider, Ph.D, Chair

Weifu Fang, Ph.D.

Gary Ganser, Ph.D.

Harumi Hattori, Ph.D.

Xin Li, Ph.D.

Department of Mathematics

Morgantown, West Virginia

2006

Keywords: Wavelets, Image Compression, Gibbs Phenomenon, Jump Discontinuity,  
Edges, Smoothing Transform, Radon Transform.

Copyright 2006, Muhammad Aslam

## ABSTRACT

### Some New Methods for Image Compression

Muhammad Aslam

The vanishing moment of wavelets and associated multi-resolution framework yield an efficient representation of smooth functions with wavelet approximation. However as pointed out in many papers, the similar result cannot be expected for the piecewise smooth functions with large jumps. In other words, wavelet approximation cannot achieve the same approximation order in the vicinity of jumps as in the smooth regions. Large wavelet coefficients associated with jumps are generated and consequently produce oscillations near discontinuities in the reconstructed signal. This is the so-called Gibbs phenomenon or a Ringing effect. In this thesis, we develop a technique which reduces the Gibbs phenomenon and improves the approximation accuracy significantly near discontinuities. We first review the wavelet transform and its responses to the typical signals. We then find an invertible smoothing transform, which modifies the signal locally in the vicinity of jump discontinuities present in the signal. Smoothing transform is applied to the signal followed by the wavelet transform. This idea can be applied recursively at several decomposition levels. In the transformed domain unimportant data is removed and the signal is reconstructed by the inverse wavelet transform followed by the inverse smoothing at each level. Our approach effectively reduces about  $\frac{1}{3}$  of the wavelet coefficients required to approximate the signal. We also give the strategy to optimize the smoothing transform. Several numerical results are presented to show the effectiveness of our algorithm. Finally, we give a new discretization of the Radon Transform and use it as an alternate method of edge detection.

This research work was partially supported by the National Science Foundation Grant NSF0314742.

# Dedication

This dissertation is dedicated  
to my wife, Farzana,  
to my sons, Umar, Abdullah and Ahmad,  
and to my parents.

## Acknowledgments

First and foremost, I want to express my sincere appreciation to my Ph.D. dissertation advisor, Dr. Sherman D. Riemenschneider for his excellent guidance, advice and continuous encouragement. It is my great pleasure to work under his supervision. Without his support, this piece of work would have never come to completion.

I would like to thank other committee members: Dr. Weifu Fang, Dr. Gary Ganser, Dr. Harumi Hattori and Dr. Xin Li for their support and help during my studies here in West Virginia University.

Many thanks to the Department of Mathematics, West Virginia University for providing me with an excellent study environment during my stay here as a graduate student. My thanks also goes to Dr. Lixin Shen for his suggestions and discussions during this research work. I would also like to thank all the professors who have given me some kind of support and encouragement during my stay in the West Virginia University.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Application of Wavelet Transform to Image Processing</b>	<b>5</b>
2.1	Multiresolution Analysis . . . . .	6
2.1.1	Series Expansion . . . . .	6
2.1.2	Multiresolution Analysis . . . . .	6
2.1.3	Wavelet Functions . . . . .	7
2.1.4	Two Dimensional Wavelets and MRA . . . . .	9
2.2	Wavelet Transform and its Response to Typical Signals . . . . .	10
2.2.1	Decomposition and Reconstruction of Discrete Signals . . . . .	10
2.2.2	Signal Approximation . . . . .	16
2.2.3	Wavelet Responses to typical signals . . . . .	22

<b>3</b>	<b>A New Approach to Image Compression</b>	<b>25</b>
3.1	Smoothing Signals . . . . .	25
3.1.1	The Modifying Transform . . . . .	26
3.1.2	Signal Modification . . . . .	28
3.2	Sparse Representation . . . . .	29
3.2.1	Algorithm for 1-D Signals . . . . .	31
3.2.2	Algorithm for 2-D Signals . . . . .	32
3.3	Optimizing our Transform . . . . .	41
3.3.1	Special Case for Daubechies wavelet filter of length 4 . . . . .	41
3.3.2	Special Case for Daubechies wavelet filter of length 6 . . . . .	45
3.4	Smoothing Transform of Higher Order . . . . .	49
3.5	Approximation Error for Piecewise Smooth Function . . . . .	51
3.6	Numerical Results . . . . .	53
3.6.1	Daubechies Wavelet Filter of length 4 is used for the following results:	53
3.6.2	Comparison: New Approach with Daubechies Wavelet Filter of length 4 with standard Bi-orthogonal Wavelet Filter of length 9 . . . . .	58
<b>4</b>	<b>Radon Transform and its Applications</b>	<b>61</b>
4.1	Radon Transform . . . . .	61

<i>CONTENTS</i>	vii
4.1.1 Fourier Approach to Compute Radon Transform . . . . .	62
4.1.2 Finite Radon Transform . . . . .	63
4.1.3 Our Approach to Distcretize Radon Transform . . . . .	64
4.2 Applications of Radon Transform . . . . .	66
4.2.1 Detection of Linear Singularities in the image . . . . .	66
4.2.2 Image Approximation . . . . .	70
4.3 Numerical Results . . . . .	71
<b>Appendix</b>	<b>75</b>
4.3.1 Modification of Piecewise Constant Function . . . . .	76
4.3.2 Modification of Piecewise Smooth Function . . . . .	78





# List of Figures

2.1	Scaling and Wavelet Subspaces . . . . .	8
2.2	Wavelet Decomposition of the signal at 1st level. . . . .	12
2.3	An example of Harr Wavelet Decomposition . . . . .	13
2.4	A signal is decomposed into scaling coefficients and wavelet coefficients . . . . .	13
2.5	Second level of Decomposition: Scaling Coefficients are further decomposed . . . . .	14
2.6	Second level of Decomposition: Scaling Coefficients are further decomposed . . . . .	14
2.7	A signal reconstruction . . . . .	15
2.8	Reconstructed signal. . . . .	15
2.9	Signal reconstruction using Harr wavelet . . . . .	16
2.10	Linear Approximation at 1st and 2nd level for 1-D signal . . . . .	18
2.11	Decomposition of 2-D signal i.e. Digital Image . . . . .	19
2.12	Digital image . . . . .	19

2.13	Column decomposition of the cameraman image . . . . .	20
2.14	1st level wavelet Decomposition of the cameraman image . . . . .	20
2.15	2st level wavelet Decomposition of the cameraman image . . . . .	21
2.16	Comparison of linear vis nonlinear approximation . . . . .	21
3.1	Piecewise smooth signal . . . . .	26
3.2	Signal modification at odd location . . . . .	27
3.3	Signal modification at even location . . . . .	27
3.4	Piecewise smooth signal and its 1st level wavelet decomposition . . . . .	29
3.5	Piecewise smooth signal and its 4 <sup>th</sup> level wavelet decomposition . . . . .	30
3.6	Comparison: New approach vis standard wavelet approach for 1-D signal .	34
3.7	Comparison: New approach vis standard wavelet approach for cameraman image using 2048 coefficients in each case . . . . .	35
3.8	Comparison: New approach with 1352 coefficients vis standard wavelet ap- proach with 2048 coefficients for cameraman image . . . . .	36
3.9	Comparison: New approach vis standard wavelet approach for rice image using 2048 coefficients in each case . . . . .	36
3.10	Comparison: New approach with 1352 coefficients vis standard wavelet ap- proach with 2048 coefficients for rice image . . . . .	37

3.11 Comparison: New approach vis standard wavelet approach for circles image using 1024 coefficients in each case . . . . .	37
3.12 Comparison: New approach with 1352 coefficients vis standard wavelet approach with 2048 coefficients for circles image . . . . .	38
3.13 Comparison: New approach vis standard wavelet approach for lena image using 4096 coefficients in each case . . . . .	38
3.14 Comparison: New approach with 3960 coefficients vis standard wavelet approach with 6000 coefficients for lena image . . . . .	39
3.15 Comparison: New approach vis standard wavelet approach for peppers image using 4096 coefficients in each case . . . . .	39
3.16 Comparison: New approach vis standard wavelet approach for barbara image using 4096 coefficients in each case . . . . .	40
3.17 Comparison: New approach vis standard wavelet approach for house image using 2048 coefficients in each case . . . . .	40
3.18 Implementation of Daubechies filter of length 4 on a signal with the discontinuity at odd location . . . . .	42
3.19 Graph of modified wavelet coefficients for odd location and the Daubechies wavelet filter of length 4 . . . . .	43
3.20 Implementation of Daubechies filter of length 4 on a signal with the discontinuity at even location . . . . .	44
3.21 Graph of modified wavelet coefficients for even location and the Daubechies wavelet filter of length 4 . . . . .	45

3.22	Graph of modified wavelet coefficients for odd location and the Daubechies wavelet filter of length 6 . . . . .	47
3.23	Graph of modified wavelet coefficients for even location and the Daubechies wavelet filter of length 6 . . . . .	48
3.24	Graph of PSNR values for the rice image with modification and without modification . . . . .	54
4.1	Simple image with linear singularities. . . . .	68
4.2	Detected Boundaries of the linear singularities of crosses image . . . . .	69
4.3	Extracted linear singularities . . . . .	69
4.4	Detected boundaries of noisy image . . . . .	70
4.5	Separation of Cartoon and the texture for crosses image . . . . .	72
4.6	Original cameraman image. . . . .	72
4.7	Separation of Cartoon and the texture for cameraman image . . . . .	73

# List of Tables

3.1	Comparison in PSNR for circles image with and without modification using Daubechies wavelet filter of length 4 . . . . .	55
3.2	Comparison in PSNR for rice image with and without modification using Daubechies wavelet filter of length 4 . . . . .	55
3.3	Comparison in PSNR for house image with and without modification using Daubechies wavelet filter of length 4 . . . . .	56
3.4	Comparison in PSNR for cameraman image with and without modification using Daubechies wavelet filter of length 4 . . . . .	56
3.5	Comparison in PSNR for peppers image with and without modification using Daubechies wavelet filter of length 4 . . . . .	57
3.6	Comparison in PSNR for lena image with and without modification using Daubechies wavelet filter of length 4 . . . . .	57
3.7	Comparison in PSNR for barbara image with and without modification using Daubechies wavelet filter of length 4 . . . . .	58

3.8	Comparison for cameraman image with modification using Daubechies wavelet filter of length 4 vis without modification using Bi-orthogonal filter of length 9. . . . .	59
3.9	Comparison for Rice image with modification using Daubechies wavelet filter of length 4 vis without modification using Bi-orthogonal filter of length 9. . . . .	59
3.10	Comparison for peppers image with modification using Daubechies wavelet filter of length 4 vis without modification using Bi-orthogonal filter of length 9. . . . .	60

# Chapter 1

## Introduction

Since their introduction, wavelets have been proved very powerful tool in signal processing. Wavelet based algorithms enjoy unusual popularity in applications including image processing. It has been verified that the wavelet transform provides an efficient representation in the sense of approximating given data with a small number of parameters. JPEG2000 is a glaring example of this.

It is a well known fact that wavelets can approximate smooth functions very efficiently but cannot have the same performance for piecewise smooth functions, especially for the functions having big jumps. The wavelet approximation produces oscillations in the vicinity of big jumps. Wavelet coefficients of large magnitude associated with these jumps are generated by the wavelet decomposition and in the reconstructed function the unaccounted wavelet coefficients of large magnitude are responsible for the oscillations or Gibbs Phenomenon. That is why nonlinear wavelet approximation (approximation by choosing the  $N$  largest coefficients irrespective of high or low frequency) has some advantage over linear wavelet approximation (approximation by  $N$  low frequency coefficients). But if all the jump-related coefficients are not preserved, one can observe the Gibbs Phenomenon even



in the nonlinear approximation of piecewise smooth function with big jumps.

One approach to minimize the Gibbs Phenomenon is to modify the wavelet transform so that large wavelet coefficients near the big jumps can be avoided. The adaptive lifting scheme of Claypoole, Davis, Sweldens and Baraniuk [19], lowers the order of approximation near the jumps and minimizes the oscillation effect is one way. Another approach by Donoho is to construct orthonormal basis like wedgelets [5], ridgelets [4], [6] and curvelets [7], [12] to give better representation of discontinuities.

Still another approach is to modify the signal instead of wavelet transform. T.F.Chan and H.M.Zhou in their adaptive ENO-Wavelet Transform [23], tried to solve this problem by modifying the signal near jumps. We adopt the later approach but in a different sense. Our approach to modify the signal is different from the ENO-Wavelet Transform of Chan and Zhou [23]. For a given smooth signal with a big jump discontinuity, the ENO-Wavelet approach treats both sides of the discontinuity as two separates signals while our approach tries to increase the smoothness in the vicinity of the jump.

We divide the problem in two parts. In the first part, we present two different approaches to detect the non-smooth regions (also called edges in image processing literature) in the 2-D signal. In the second part, we derive an invertible transform to modify the non-smooth parts of the signal (to make it locally  $C^{(k)}$  in some sense) in order to get a better approximation order near the edges.

In Chapter 2, we give a brief review of Wavelet Transform and its applications to discrete signals. Wavelet decomposition and reconstruction of 1-D signals and 2-D signals (images) is discussed with examples. Wavelet responses to typical signals are also explained along with some other properties.

Chapter 3 is devoted to reducing the ringing effect in the approximated signals. We

use first derivative to determine the non-smooth regions in the signal as a first part of the problem and for the second part of the problem, we develop an invertible smoothing transform. This transform adaptively modifies the signal and performs as a pre- and post-processing for the standard wavelet approximation. The signal is modified in such a way that in the wavelet decomposition almost all the energy of the modified signal concentrates in the coarse coefficients (coefficients corresponding to low pass filter). Then a nonlinear approximation approach based on thresholding is applied followed by the inverse of the modification process.

Chapter 3 also includes ways to optimize our smoothing transform for different orthogonal wavelet filters. We present several numerical results to show the performance of our approach as compared to a wavelet filter of higher order. Our approach works well for all the images normally used by the image processing people to judge results and there is greater improvement if a small number of coefficients are used to approximate the image.

At present, there does not exist an approximation error bound for the piecewise smooth functions with large jumps. However, we have proved the approximation error bound for the given piecewise smooth function  $f$  with the large jump at the point  $x$  under the assumption that it has  $p^{th}$  order bounded derivatives on the smooth regions.

In Chapter 4, we use Radon Transform as an alternate approach to detect the non-smooth regions in the given image (as a first part of the problem). Because of the Radial nature of the Radon transform, its implementation to the digital images is not simple. We first discuss two existing approaches of the implementation of Radon Transform to digital images as described in [1], [8], [10], [12], [14], [15] and [18] and the problems associated with these methods. Then we present a new discretization to the Radon Transform and use it to determine the linear singularities present in digital images.



## Chapter 2

# Application of Wavelet Transform to Image Processing

In this chapter, we have given a brief introduction of wavelet transform from the multiresolution point of view. Signal decomposition into scaling coefficients and wavelet coefficients is described with examples. Wavelet based algorithms for linear and nonlinear approximation of the signal are also discussed. Wavelet response to typical signals and Gibbs Phenomenon are explained especially for the piecewise smooth signals.

## 2.1 Multiresolution Analysis

### 2.1.1 Series Expansion

A signal or function  $f(x)$  can often be better analyzed as a linear combination of expansion functions.

$$f(x) = \sum_i \alpha_i \phi_i(x),$$

where  $i$  is the integer index of the finite or infinite sum, the  $\alpha_i$ 's are called expansion coefficients and the  $\phi_i(x)$  are expansion functions. If  $\{\phi_i(x) : i \in Z\}$  is linearly independent and span the function space to which  $f(x)$  belongs then  $\phi_i(x)$  are called basis functions and in that case  $\alpha_i$ 's are unique in the above expansion.

We are interested in some special expansion functions with scaling and translation properties. Before introducing these functions, it is imperative to define a process which provides a systematic approach for their construction.

### 2.1.2 Multiresolution Analysis

(Mallat and Meyer 1986)[21]

Multiresolution Analysis (MRA) is an elegant mathematical framework for representing and analyzing signals and images in multiple scales. An orthonormal MRA of  $L^2(R)$  is an ordered chain of closed subspaces (also known as scaling subspaces):

$$\cdots \subseteq V_{-1} \subseteq V_0 \subseteq V_1 \subseteq \cdots,$$

which satisfies the following three conditions:

- (Completeness)  $\overline{\lim_{i \rightarrow \infty} V_i} = L^2(R)$ , and  $\lim_{i \rightarrow -\infty} V_i = \{0\}$ .

- (Scale Similarity)  $u(x) \in V_j \Leftrightarrow u(2x) \in V_{j+1}$ .
- (Translation Seed) There exists a function  $\phi(x) \in V_0$ , so that  $\{\phi(x - k) : k \in Z\}$  is an orthonormal basis of  $V_0$ . The function  $\phi(x)$  is called a scaling function.

We can define

$$\phi_{j,k}(x) = 2^{j/2} \phi(2^j x - k), \quad j, k \in Z.$$

By the dyadic/scale similarity,  $\{\phi_{j,k}(x) : k \in Z\}$  is an orthonormal basis of  $V_j$ . Thus the scaling function  $\phi(x)$  can be expressed as (because  $V_0 \subset V_1$ ):

$$\phi(x) = \sum_k h[k] \phi_{1,k}(x) = \sqrt{2} \sum_k h[k] \phi(2x - k).$$

The sequence  $\{h[k] = \int \phi(x) \phi_{1,k}(x) dx\}$  is called the **lowpass filter**.

By the completeness property of MRA, any function  $f(x) \in L^2(R)$  can be well approximated to any desired precision by its projection  $f_j(x) = P_j f(x)$  onto  $V_j$ :

$$f_j(x) = \sum_k \alpha(k) \phi_{j,k}(x)$$

### 2.1.3 Wavelet Functions

Let  $W_0$  denote the orthogonal complement of  $V_0$  in  $V_1$ , i.e.  $W_0 = V_1 \cap V_0^\perp$  or  $V_1 = V_0 \oplus W_0$ . Then  $W_0$  is also orthogonally spanned by the integer translates of a single translation seed  $\psi(x)$ , called a **wavelet**:

$$\psi(x) = \sum_k g[k] \phi_{1,k}(x) = \sqrt{2} \sum_k g[k] \phi(2x - k).$$

The sequence  $\{g[k] = \int \psi(x) \phi_{1,k}(x) dx\}$  is called the **highpass filter**.

We define the wavelet subspaces:  $W_j = V_{j+1} \cap V_j^\perp$  and the set  $\{\psi_{j,k}(x)\}$  of wavelets:

$$\psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k) \quad j, k \in Z.$$

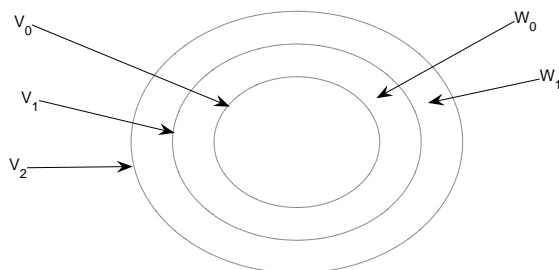


Figure 2.1: Scaling and Wavelet Subspaces

Because of the orthogonality of the subspaces  $W_j$ 's, we have

$$L^2(\mathbb{R}) = V_J \oplus \sum_{j \geq J} W_j = \sum_{j=-\infty}^{\infty} W_j$$

and, therefore,  $\{\psi_{j,k}(x) : j, k \in \mathbb{Z}\}$  form an orthonormal basis of  $L^2(\mathbb{R})$ .

Any  $f \in L^2(\mathbb{R})$  can be expressed as

$$f(x) = \sum_k \alpha_J(k) \phi_{J,k}(x) + \sum_{j \geq J} \sum_k \beta_j(k) \psi_{j,k}(x),$$

where

$$\alpha_J(k) = \int f(x) \phi_{J,k}(x) dx$$

and

$$\beta_j(k) = \int f(x) \psi_{j,k}(x) dx.$$

The  $\alpha_J(k)$  are called scaling coefficients at  $J^{\text{th}}$  level and the  $\beta_j(k)$  are called wavelet coefficients at  $j^{\text{th}}$  level.

### 2.1.4 Two Dimensional Wavelets and MRA

The concept of a one dimensional MRA can easily be extended to the two dimensional MRA in the sense of tensor products [22]. The space  $V_{(j)} = V_j \otimes V_j$  and  $W_{(j)} = V_{(j+1)} \cap V_{(j)}^\perp$ . Using the separability condition of tensor Product, all the conditions of MRA follow.

Define

- $\Phi(x, y) = \phi(x)\phi(y)$  Scaling function
- $\Psi^H(x, y) = \phi(x)\psi(y)$  Horizontal sensitive wavelet
- $\Psi^V(x, y) = \psi(x)\phi(y)$  Vertical sensitive wavelet
- $\Psi^D(x, y) = \psi(x)\psi(y)$  Diagonal sensitive wavelet

Let  $\Phi_{j,(k_1,k_2)}(x, y) = 2^j \Phi(2^j x - k_1, 2^j y - k_2)$  and

$$\Psi_{j,(k_1,k_2)}^\alpha(x, y) = 2^j \Psi^\alpha(2^j x - k_1, 2^j y - k_2) \text{ for } \alpha \in \{H, V, D\}.$$

Then

$\{\Phi_{j,(k_1,k_2)}(x, y) : k_1, k_2 \in Z\}$  forms an orthonormal basis for  $V_{(j)}$  and

$\{\Psi_{j,(k_1,k_2)}^\gamma(x, y) : k_1, k_2 \in Z, \gamma = H, V, D\}$  forms an orthonormal basis for  $W_{(j)}$ .

Therefore, any  $f \in L^2(R^2)$  is given by [20];

$$f(x, y) = \sum_{k_1, k_2} \alpha_J(k_1, k_2) \Phi_{J,(k_1,k_2)}(x, y) + \sum_{j \geq J} \sum_{k_1, k_2} \sum_{\gamma} \beta_j(k_1, k_2) \Psi_{j,(k_1,k_2)}^\gamma(x, y).$$



## 2.2 Wavelet Transform and its Response to Typical Signals

In this section we discuss how the wavelet transform decomposes the discrete signal into scaling and wavelet coefficients and vice versa. We also give brief description of wavelet responses to smooth and non-smooth signals.

### 2.2.1 Decomposition and Reconstruction of Discrete Signals

#### One Dimensional Signal Decomposition:

Let  $X(n)$  be a given discrete signal and  $h(n)$  and  $g(n)$  be the lowpass and highpass filters respectively (as discussed in the previous section), obtained from the specific wavelet used to decompose the signal. One-dimensional (1-D) wavelet transform can be implemented by convolving the lowpass and highpass filters with the given signal. The convolution of the signal with the lowpass filter will give scaling coefficients and the convolution with highpass filter will give wavelet coefficients. Mathematically, the forward discrete wavelet transform is expressed by [2], [3] and [11],

$$s_{-1}[\ell] = \sum_n s_0[n]h[n - 2\ell], \quad d_{-1}[\ell] = \sum_n s_0[n]g[n - 2\ell]$$

where  $s_0[n] = X(n)$ ,  $s_{-1}[\ell]$  are the scaling coefficients and  $d_{-1}[\ell]$  are the wavelet coefficients.

Once the signal is decomposed, it is down-sampled by the factor 2, because on decomposition the data will double and half of it is redundant. Therefore, every other scaling and wavelet coefficient is removed. The graphic explanation is given in the figure 2.2, figure 2.3 and the figure 2.4.

Depending upon the application, scaling and/or wavelet coefficients can be further decomposed into the scaling and wavelet coefficients at the next level and so on. Figure 2.5 and figure 2.6 explain further decomposition of scaling coefficients.

### Two Dimensional Signal Decomposition:

Like the 1-D wavelet transform, the 2-D wavelet transform can be implemented using filters and the down-sampling. We simply take 1-D wavelet transform of the columns of an image (i.e. 2-D signal), followed by the 1-D wavelet transform of the resulting rows. Figure 2.11 shows the process in block diagram form. Note that, like its 1-D counterpart in figure 2.2, the 2-D wavelet transform generates scaling and wavelet coefficients. In the 2-D case, however, we get three sets of wavelet coefficients namely horizontal, vertical and diagonal wavelet coefficients. The wavelet decomposition of an image as described in figure 2.11 can be iterated i.e. the scaling coefficients can be further decomposed into scaling and wavelet coefficients at the next level and so on.

Convolving the columns of an image with  $h(n)$  and  $g(n)$  and down-sampling by 2, we get two sub-images whose vertical resolutions are reduced by the factor 2. Figure 2.13 contains two sub-images (just attached to each other) obtained by the above process from the image given in figure 2.12. The lowpass output characterizes the low-frequency vertical information while the highpass output contains high-frequency vertical information. Rows of the both sub-images are then convolved with  $h(m)$  and  $g(m)$  and down-sampled to yield four quarter size sub-images. One sub-image will consist of scaling coefficients while the other three will be of horizontal, vertical and diagonal wavelet coefficients. Figure 2.14 contains four such sub-images of the image given in the figure 2.12. Similarly, Two iterations (i.e. decomposing the scaling coefficients further) will produce eight sub-images. Second level of decomposition of the image given in figure 2.12 is described in figure 2.15.

### Signal Reconstruction:

The original signal can be exactly reconstructed from the scaling coefficients and the wavelet coefficients by using the inverse discrete wavelet transform. The procedure is opposite to the decomposition. First the coefficients are up-sampled by the factor 2 by

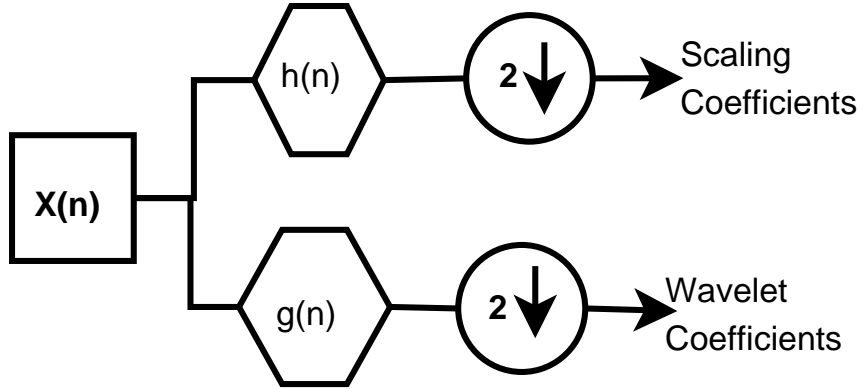


Figure 2.2: Wavelet Decomposition of the signal at 1st level.

adding 0's and then inverse filters  $h'[n] = h[-n]$  and  $g'[n] = g[-n]$  are convolved with the up-sampled coefficients and the result is added, as described in the figure 2.7, figure 2.8 and the figure 2.9. If  $s_{j-1}[\ell]$  and  $d_{j-1}[\ell]$  are the scaling coefficients and the wavelet coefficients respectively at the  $j - 1$  level of decomposition, then the scaling coefficients  $s_j[n]$  at the previous level can be found in [2], [3] and [11] by the expression,

$$s_j[n] = \sum_{\ell} s_{j-1}[\ell] h'[n - 2\ell] + \sum_{\ell} d_{j-1}[\ell] g'[n - 2\ell].$$

This procedure is repeated at each level until we get the original signal.

In case of 2-D signal, reconstruction procedure is similar to the 1-D case. At each level the rows of the four sub-images are up-sampled and then convolved with the inverse filters followed by the up-sampling of the resulting columns and their convolution with inverse filters. The resulting image will be the image of scaling coefficients at the previous level. Continue this process until, we get the original image.

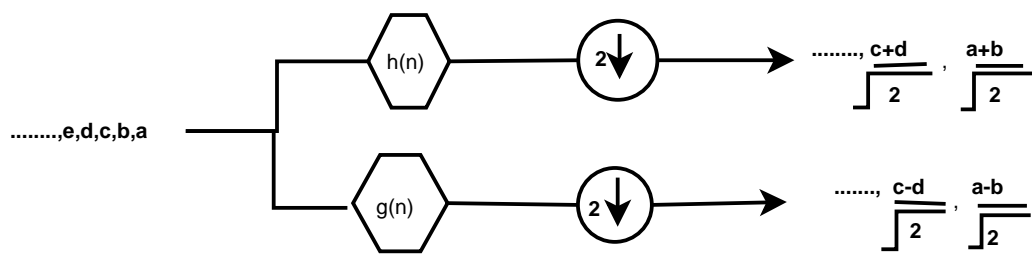


Figure 2.3: An example of Harr Wavelet Decomposition, where  $h = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$  and  $g = [\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]$ .

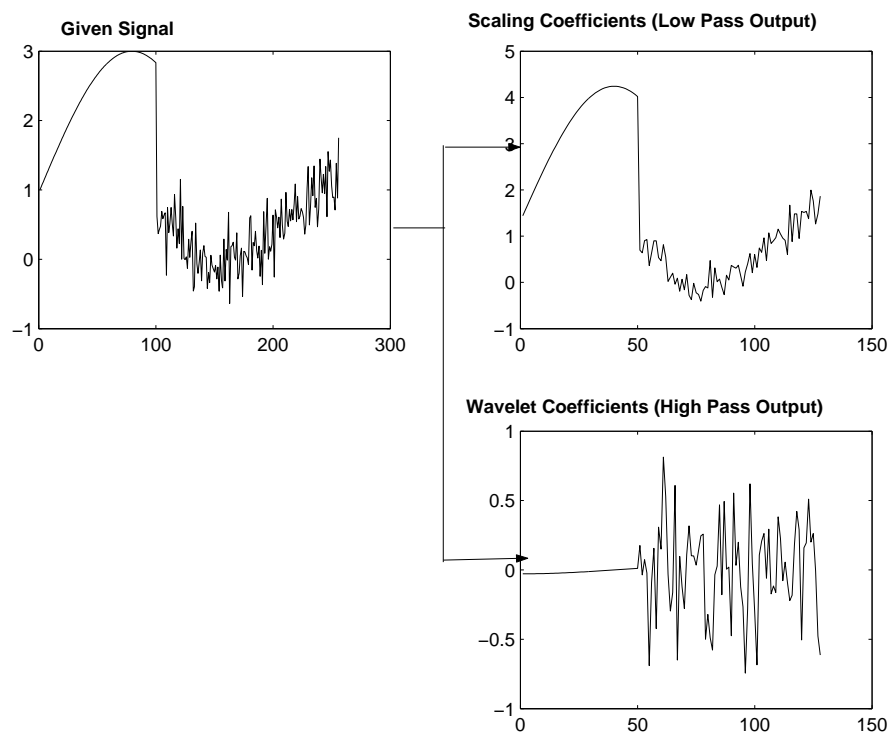


Figure 2.4: A signal is decomposed into scaling coefficients and wavelet coefficients

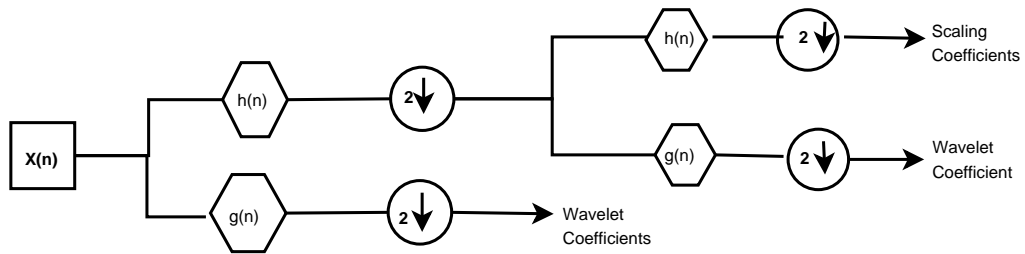


Figure 2.5: Second level of Decomposition: Scaling Coefficients are further decomposed

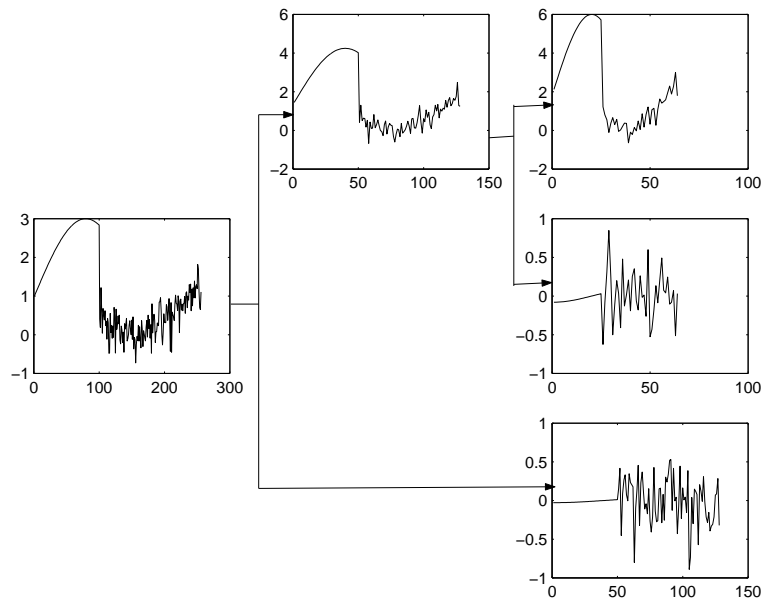


Figure 2.6: Second level of Decomposition: Scaling Coefficients are further decomposed

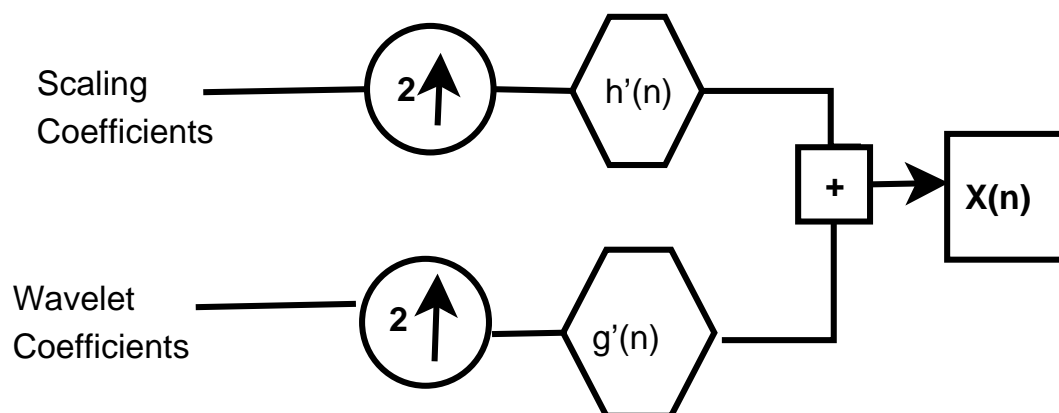


Figure 2.7: A signal reconstruction from the scaling coefficients and the wavelet coefficients, where  $h'(n) = h(-n)$  and  $g'(n) = g(-n)$ .

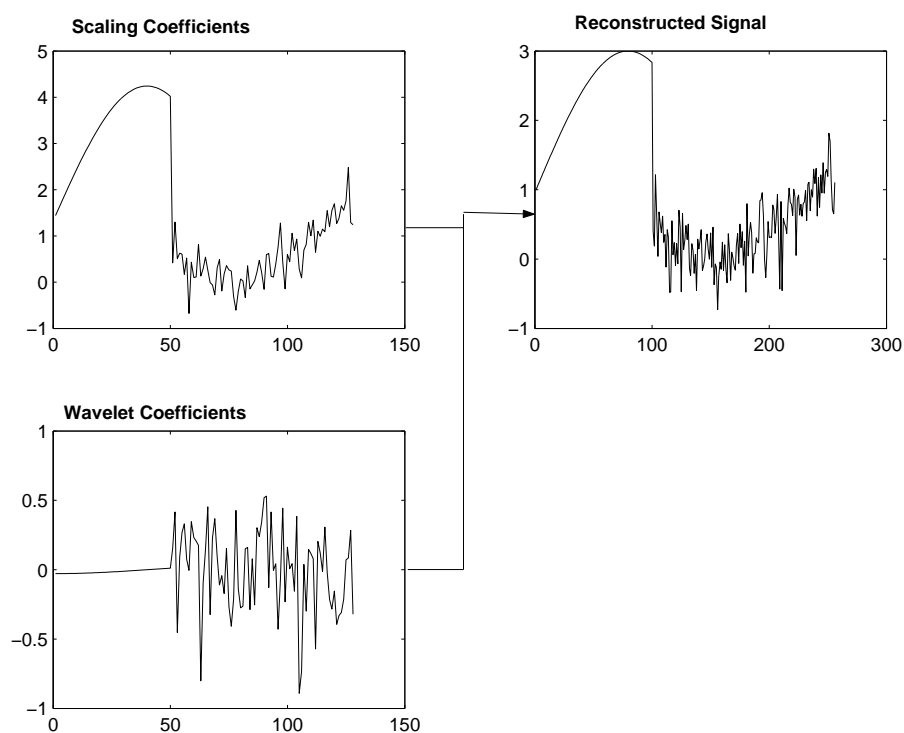


Figure 2.8: Reconstructed signal.

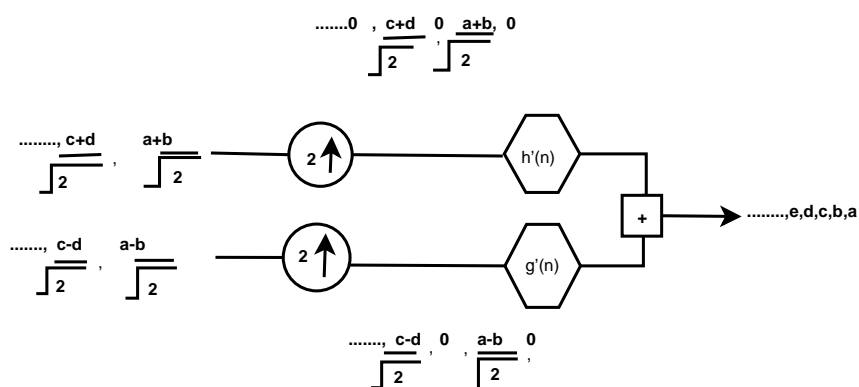


Figure 2.9: An example of Harr wavelet, where  $h' = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$  and  $g' = [-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$ .

### 2.2.2 Signal Approximation

In image compression applications, the main objective is to reduce the amount of data to represent the original signal or its approximation. Therefore, we try to get a better approximation of the signal while using the small amount of data. We use the Wavelet transform to reconstruct the approximation of the signal. The procedure for wavelet approximation has the following three main steps.

1. A signal is decomposed using the forward wavelet transform into scaling and wavelet coefficients.
2. Some kind of alteration is made in the coefficients. This alternation may be in the form of deleting some unimportant coefficients.
3. Inverse wavelet transform is applied to altered set of coefficients and the result will be the approximation of the original signal.

### Wavelet Linear Approximation

If in the second step of the wavelet approximation's procedure, we alter the coefficients in such a way that all the wavelet coefficients become zero and the scaling coefficients remain unchanged. Then the resulting approximation of the signal will be called linear wavelet approximation. In other words, wavelet linear approximation means the reconstruction of the signal only from the scaling coefficients or from the lowpass output, while setting all the wavelet coefficients or highpass Output equal to zero.

Wavelet approximation resulting from the scaling coefficients at  $j^{th}$  level of decomposition is called wavelet linear approximation at  $j^{th}$  level or the  $j^{th}$  level wavelet approximation. Figure 2.10 explains the wavelet linear approximation at 1st and 2nd level.

### Wavelet Nonlinear Approximation

If in the second step of wavelet approximation's procedure, we decide to keep only the coefficients of significant magnitude and put all the other coefficients equal to zero. This can be accomplished by hard thresholding, which means setting to zero the coefficients whose absolute values are lower than the threshold. Then the resulting approximation will be called wavelet nonlinear approximation.

In other words, wavelet nonlinear approximation means the approximation of a signal by the  $N$  largest coefficients (irrespective of scaling or wavelet) in magnitude, while setting all the remaining coefficients equal to zero. Figure 2.16 shows the comparison of wavelet linear and nonlinear approximation.

### Gibbs Phenomenon:

Approximating the signal only by the lowpass output or scaling coefficients causes oscillations near the large jumps discontinuities in the signal [21]. This is called Gibbs



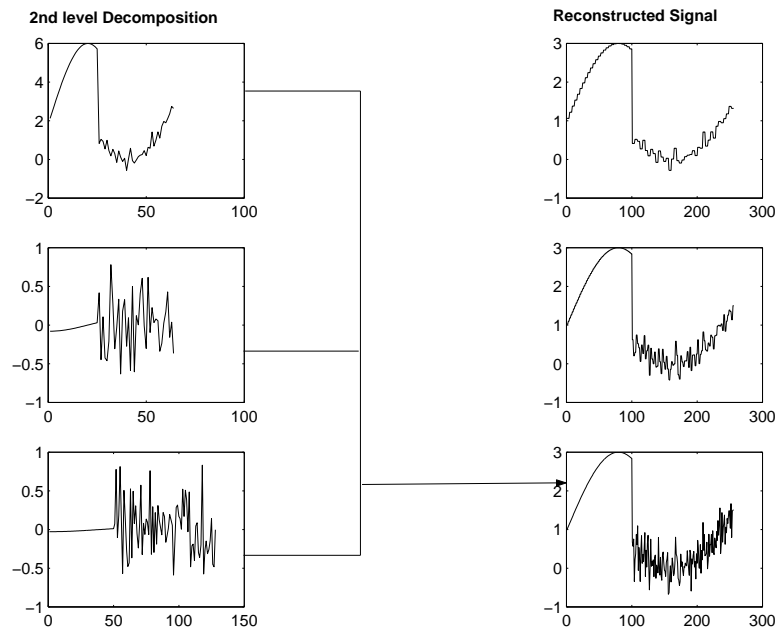


Figure 2.10: On the left column are the scaling and wavelet coefficients at second level of decomposition, on the right top is the reconstructed signal only from scaling coefficients (i.e. wavelet linear approximation at 2nd level), on the middle of the second column is reconstructed signal from scaling coefficients and some wavelet coefficients (wavelet linear approximation at 1st level) and in the lower second column is the reconstructed signal from all the scaling and wavelet coefficients.

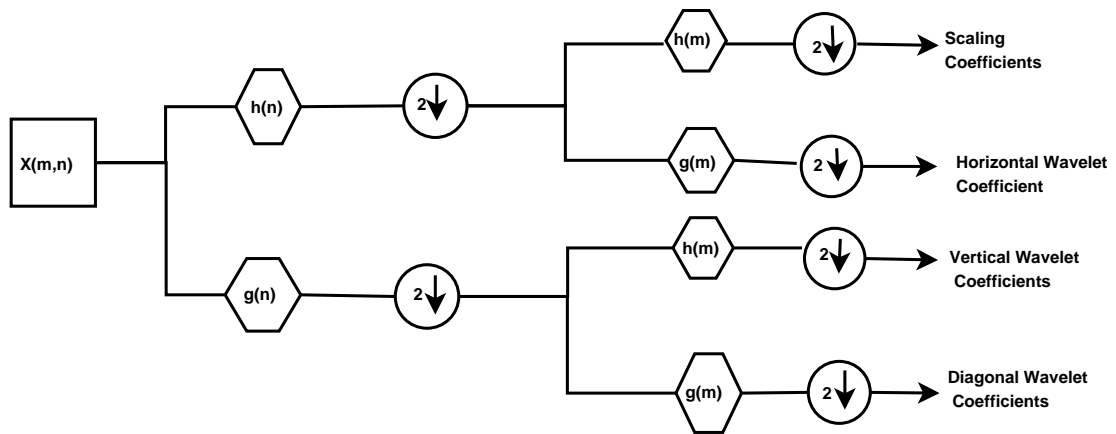


Figure 2.11: Decomposition of 2-D signal i.e. Digital Image



Figure 2.12: Digital image



Figure 2.13: Wavelet Transform is applied along columns: Upper part of the image is the output of lowpass filter and lower part is the output of the highpass filter.



Figure 2.14: Wavelet Decomposition at 1st level produces 4 sub-images: On the upper left corner is an image of scaling coefficients, on the lower left corner is an image of vertical wavelet coefficients, on the upper right corner is an image of horizontal wavelet coefficients and on the lower right is an image of diagonal wavelet coefficients.

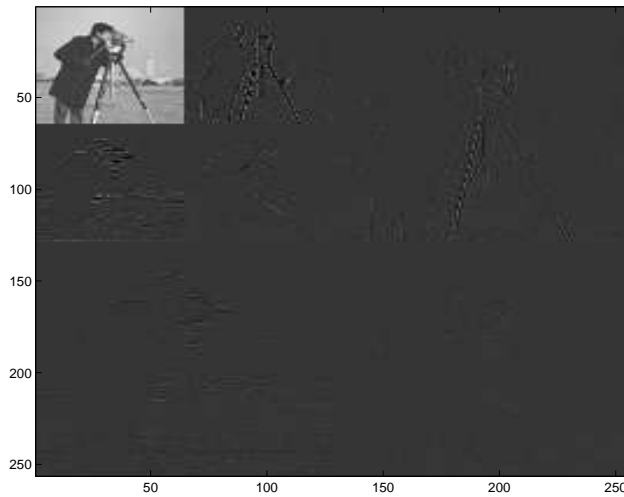


Figure 2.15: Wavelet Decomposition at 2nd level. Scaling coefficients are further decomposed.



Figure 2.16: On the left side image approximated by linear approach while on the right side by nonlinear approach: Same number of coefficients (1024) are used in both cases

Phenomenon or Gibbs Oscillations. In image processing it is also called the Ringing Effect.

The Gibbs Phenomenon may also occur in wavelet nonlinear approximation if the number of coefficients used to approximate the signal is small.

### 2.2.3 Wavelet Responses to typical signals

A given  $f(x) \in L^2(\mathbb{R})$  can be linearly approximated by its projection  $f_j(x) \in V_j$  as described in the MRA section. This linear approximation has a standard error estimates, which can be described in the following theorem; see also [9] and [23].

**Theorem 2.2.1** *Suppose the wavelet  $\psi(x)$  generated by the scaling function  $\phi(x)$  has  $p$  vanishing moments and  $f_j(x)$  is the approximation of  $f(x)$ , which has bounded  $p^{\text{th}}$  order derivative, in  $V_j$  with basis  $\phi_{j,k}(x)$ ; then,*

$$\|f(x) - f_j(x)\|_2 \leq C(\Delta x)^p \|f^{(p)}(x)\|_2$$

where  $\Delta x = 2^{-j}$  and  $C$  is a constant independent of  $j$ .

This theorem holds for  $L^2$  norm in general. Moreover, if the scaling function and its wavelet have finite support, then it also holds for the  $L^\infty$  norm.

However if the function has a discontinuity in its  $m^{\text{th}}$  derivative (with  $m \leq p$ ) then the above result does not hold and we don't have an error bound in that case [23]. In the following, we discuss what happens to functions with discontinuities.

**Definition 2.2.2** *A function  $f(x)$  has a jump discontinuity in its  $m^{\text{th}}$  derivative at a point  $x_0$  if the left and the right limits of the derivatives  $f^{(m)}(x_0)$  are not equal; i.e.*

$$f^{(m)}(x_0-) \neq f^{(m)}(x_0+).$$

We define the intensity of a jump in the  $m^{\text{th}}$  derivative at  $x_0$  as

$$J_m f(x_0) = |f^{(m)}(x_0+) - f^{(m)}(x_0-)|.$$

Using Taylor series in the small neighborhood of  $x_0$ , we can write such a function  $f(x)$  as

$$f(x) = g(x) + \begin{cases} f^{(m)}(x_0-)(x - x_0)^m + O(x - x_0)^{(m+1)} & \text{if } x \leq x_0 \\ f^{(m)}(x_0+)(x - x_0)^m + O(x - x_0)^{(m+1)} & x_0 < x, \end{cases}$$

where  $g(x)$  is a Taylor polynomial of  $f(x)$  of order  $m - 1$  near  $x_0$ .

If we use a wavelet filter of length  $l$  and assume that  $x_0 \in (i\Delta x, (i + l)\Delta x)$  then the wavelet coefficient (or output of the highpass filter) at the  $j^{\text{th}}$  level of wavelet decomposition  $\beta_{j,i}$  as given by [23];

$$\begin{aligned} |\beta_{j,i}| &= \left| \int f(x) \psi_{j,i}(x) dx \right| \\ &= \left| \int_{i\Delta x}^{x_0} (f^{(m)}(x_0-)(x - x_0)^m + O(x - x_0)^{(m+1)}) \psi_{j,i}(x) dx \right. \\ &\quad \left. + \int_{x_0}^{(i+l)\Delta x} (f^{(m)}(x_0+)(x - x_0)^m + O(x - x_0)^{(m+1)}) \psi_{j,i}(x) dx \right|, \end{aligned}$$

which implies

$$|\beta_{j,i}| = |J_m f(x_0)| O((\Delta x)^m).$$

### Summary of the Above Discussion:

- In a wavelet decomposition of a smooth signal there is no **Wavelet Coefficient** of significant magnitude.
- If the signal has a jump discontinuity then the wavelet decomposition may generate **Wavelet Coefficients** of significant magnitude and their magnitude is proportional to the size of the jump discontinuity.

- Unaccounted Significant Wavelet Coefficients are responsible for the **Gibbs Phenomenon or Approximation Error**. Here unaccounted coefficients mean those coefficients which we put equal to zero in the second step of the procedure of wavelet approximation.

## Chapter 3

# A New Approach to Image Compression

In image compression, the aim is to reduce the amount of data required to represent a given digital image. The underlying basis of the reduction process is the removal of redundant data. Some transform is applied for this purpose and the remaining data is compressed for storage or transmission. At some later time, compressed data is decompressed to reconstruct an original image or its approximation.

### 3.1 Smoothing Signals

In Chapter 1, we have discussed how the jump in the signal produces wavelet coefficients of significant magnitude and consequently affects the approximation of the signal. In order to avoid the production of wavelet coefficients of significant magnitude caused by jumps, we can either modify the wavelet transform or modify the signal locally. The idea for the



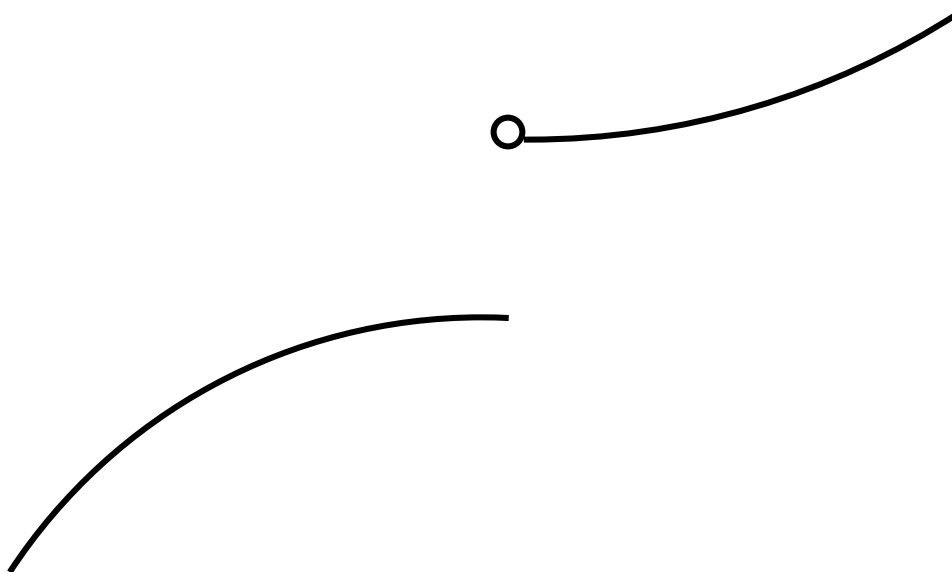


Figure 3.1: Piecewise smooth signal

second approach is to apply the wavelet transform on relatively smoother data, and thus achieve better approximation.

### 3.1.1 The Modifying Transform

For our discussion, we assume that the signal is piecewise smooth and any two discontinuities are at least  $L - 1$  pixels apart, where  $L$  is the length of the wavelet filter to be used.

Define the matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ -a & 1 & a \\ 0 & 0 & 1 \end{pmatrix}$$

where  $a \in (0, 1)$ . The value of  $a$  will depend upon the wavelet filter used, and the

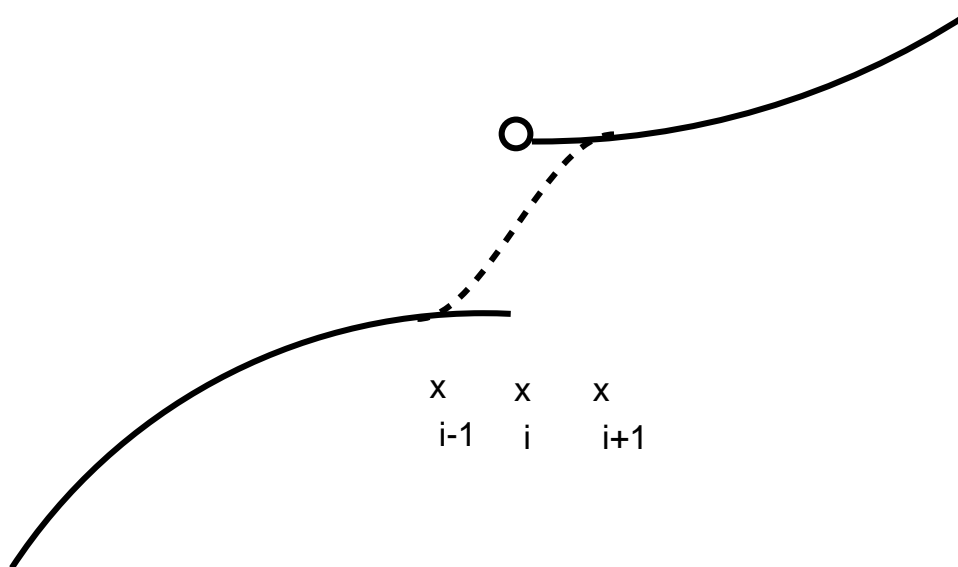


Figure 3.2: A Piecewise smooth signal given in figure 3.1 is modified when  $i = \text{odd}$ .

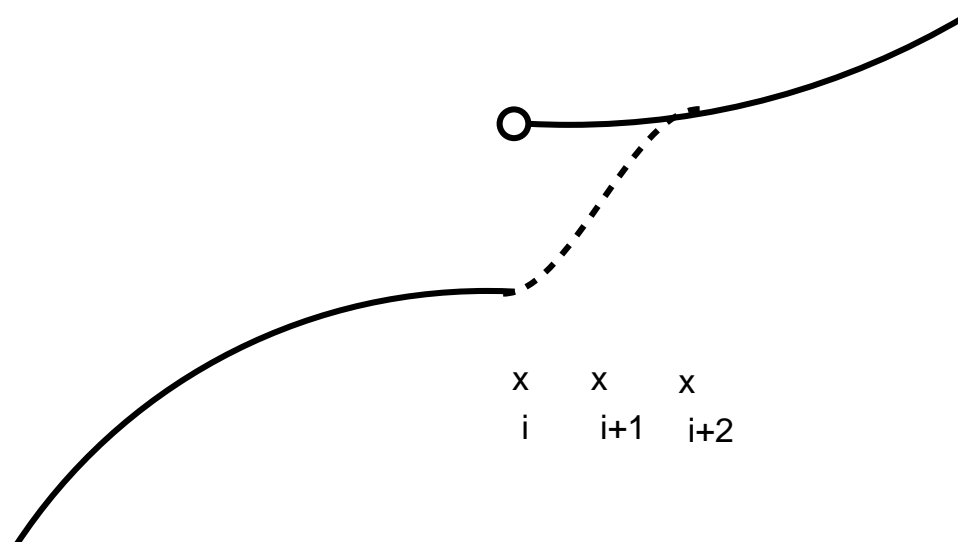


Figure 3.3: A Piecewise smooth signal given in figure 3.1 is modified when  $i = \text{even}$ .

even/odd location of the jump discontinuity in the sequence of data.

Note that  $\mathbf{A}$  is invertible with inverse

$$\mathbf{A}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ a & 1 & -a \\ 0 & 0 & 1 \end{pmatrix}$$

### 3.1.2 Signal Modification

1. Let  $\mathbf{f} = \{f_i\}$  be a discrete signal having a big jump discontinuity at the  $i^{\text{th}}$  step. By that we mean  $|f_i - f_{i+1}| > T$  for some prescribed threshold  $T$ . In order to modify the signal we can either modify the value  $f_i$  or the value  $f_{i+1}$ .
2. **Modifying  $f_i$ :** Apply the **matrix  $A$  on the segment**  $\{f_{i-1}, f_i, f_{i+1}\}$  of the signal to obtain the modified value  $\tilde{f}_i = f_i + a(f_{i+1} - f_{i-1})$ . A continuous counterpart signal of this discrete modification is shown in the figure 3.2.
3. **Modifying  $f_{i+1}$ :** Apply the **inverse of matrix  $A$  on the segment**  $\{f_i, f_{i+1}, f_{i+2}\}$  of the signal to obtain the modified value  $\tilde{f}_{i+1} = f_{i+1} - a(f_{i+2} - f_i)$ . A continuous counterpart signal of this discrete modification is shown in the figure 3.3.

**Theorem 3.1.1** *Let  $f$  be a given piecewise smooth signal having big jump discontinuities and  $\tilde{f}$  be the modified signal by using the above procedure on the points of discontinuities. Then the modified signal  $\tilde{f}$  will be smoother than  $f$  in the sense that*

$$\sum_j |\Delta^2 \tilde{f}_j|^2 < \sum_j |\Delta^2 f_j|^2.$$

**Proof** The proof of this theorem can be found in appendix.

Figure 3.4 shows the piecewise smooth signal with one jump and its wavelet decomposition at 1st level by using Harr wavelet. One can see the significant wavelet coefficient

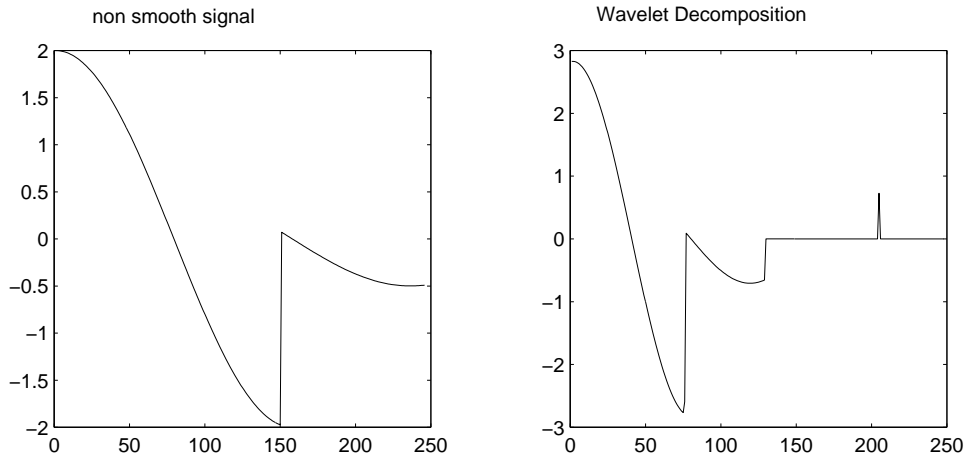


Figure 3.4: On the left, Piecewise smooth signal with one discontinuity and on the right its wavelet decomposition at 1st level. Wavelet Coefficient of significant magnitude is visible corresponding to the discontinuity.

corresponding to the jump discontinuity. Figure 3.5 shows the 4<sup>th</sup> level of wavelet decomposition of the same piecewise smooth signal with modification and without modification by using Daubechies wavelet filter of length 4. One can see the advantage of modification that the number of significant wavelet coefficients in the wavelet decomposition of modified signal are less compared to the number of significant wavelet coefficients in the wavelet decomposition of the signal without modification.

## 3.2 Sparse Representation

In this section, we try to improve the quality of the approximated image without increasing the amount of data to represent it. We use modification and inverse modification process in addition to the standard wavelet nonlinear approximation. Our algorithms given in this section are for the piecewise smooth signals with big jump discontinuities.

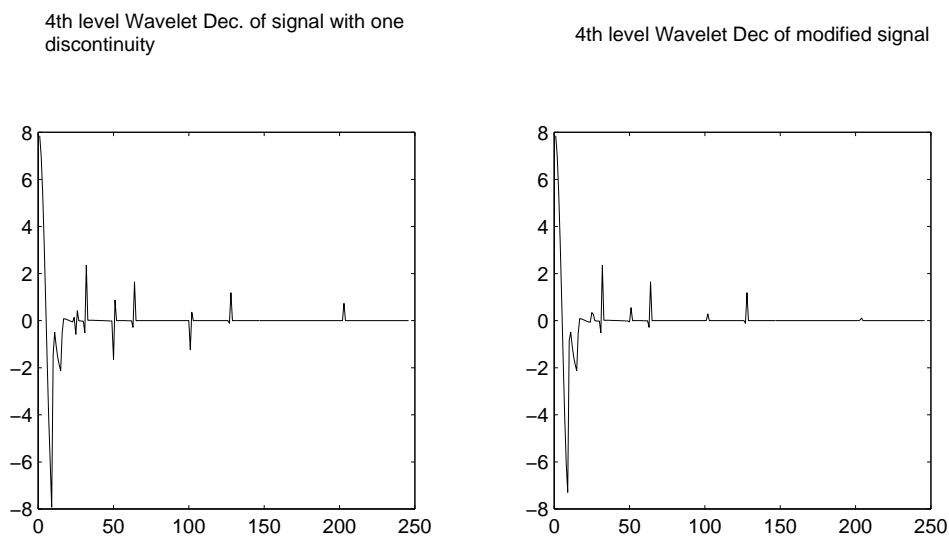


Figure 3.5: Daubechies wavelet filter of length 4 is used to decompose the signal at 4th level. One can see that the number of significant wavelet coefficients are less in the decomposition of modified signal.

### 3.2.1 Algorithm for 1-D Signals

Let  $\mathbf{f} = \{f_i\}$  be a discrete 1-D signal. We assume that these discrete values are from a piecewise smooth function having big jump discontinuities. We further assume that any two jump discontinuities are at least  $L - 1$  pixels apart, where  $L$  is the length of the wavelet filter to be used. Our algorithm for sparse representation of the 1-D signal can be expressed in the following steps:

#### 1. Jump Detection

If  $|f_i - f_{i+1}| > T_1$ , where  $T_1$  is some threshold. Then the function  $\mathbf{f} = \{f_i\}$  has a big jump at the  $i^{\text{th}}$  step.

#### 2. Modification

If this jump generates a wavelet coefficients of significant magnitude i.e.  $|\beta_{j,i}| > T_2$ , where  $T_2$  is some other threshold. Then we apply the transform  $A$  or its inverse to modify the signal locally.

#### 3. Decomposition

Decompose the modified signal by the wavelet transform and apply the above idea recursively at different levels of decomposition in the multiresolution framework.

#### 4. Alteration

Set all the coefficients except  $N$  largest coefficients equal to zero.

#### 5. Reconstruction

Use inverse wavelet transform on the altered set of coefficients.

#### 6. Inverse Modification

Apply the inverse of the modifying process locally after the reconstruction step.

Above two steps are recursively applied at each level if the signal is decomposed at multi-level.

### 3.2.2 Algorithm for 2-D Signals

Let  $X$  be an image(2-D signal) of size  $m$  by  $n$ . The 2-D implementation of the above algorithm is very simple. Here are the steps to explain the 2-D implementation of this algorithm.

1. We go first along column by column and detect the big jump singularities. Apply the transform  $\mathbf{A}$  or  $\mathbf{A}^{-1}$  on the appropriate segments of the columns corresponding to the singularities if these jump singularities are responsible for the generation of significant wavelet coefficients.
2. Find a wavelet decomposition of columns. This will result in two subimages, one corresponding to the output of the lowpass filter and the other corresponding to the output of the highpass filter.
3. Detect the jump in each row of the subimage corresponding to the output of the lowpass filter. Modify the jumps by  $\mathbf{A}$  or  $\mathbf{A}^{-1}$  on the appropriate segments of the rows, which are responsible for the generation of significant wavelet coefficients.
4. Find a wavelet decomposition of rows of the subimages. As a result, we have four subimages.
5. The above steps may be repeated for further smoothing and decomposition of the output of the lowpass filter.
6. After the wavelet decomposition of the image at certain levels,  $N$  largest coefficients in magnitude are selected and all the remaining coefficients are set equal to zero.
7. From the altered set of coefficients, image is reconstructed step by step, by the inverse wavelet transform followed by the inverse modification process. That is, at each level, inverse wavelet transform is applied followed by the reverse of the smoothing process.

### Detection of Jump Singularities

There are many ways to detect the big jumps, we choose to compare the intensity values of the neighboring pixels to find the big jumps. Let  $x = X(:, k)$  be the  $k^{\text{th}}$  column which is under inspection for finding jump singularities. If for adjacent pixel values,  $x_{i-1}$ ,  $x_i$ , in the column, we have  $|x_{i-1} - x_i| > T_1$  for some threshold  $T_1 > 0$ , then we consider that a big jump occurs between the values  $x_{i-1}$  and  $x_i$ . Once the big jump is detected it is further investigated for its role in generating significant wavelet coefficients. If it is responsible for producing any wavelet coefficient of significant magnitude, then we apply the transform  $\mathbf{A}$  or  $\mathbf{A}^{-1}$  on the appropriate segment of  $x$  to break the big jump into smaller jumps and so on. A similar procedure will be followed for each row.

However, if the signal or the image is noisy, this procedure may fail to find the big jumps or may find the jump at the wrong position. For such problems one can use other methods for singularity detection [23].

### Selection of Threshold

Selection of the threshold is always a difficult part of the problem. We have adopted a simple strategy for this. From our discussion in the previous sections, we know that magnitude of the wavelet coefficient generated by the jump is proportional to the intensity of that jump and our aim is to stop the generation of wavelet coefficients of big magnitude. Therefore, we only need to find the position of big jumps and consequently we need a larger value of the Threshold  $T_1$ . In our experiments, we have used the standard deviation of the signal as a threshold  $T_1$  for detecting the big jumps.

Significance of the wavelet coefficients depends upon the number  $N$  of the coefficients, which are kept during the alteration step of the algorithm. A wavelet coefficient which is significant for a large value of  $N$  may become insignificant for the smaller value of  $N$ .



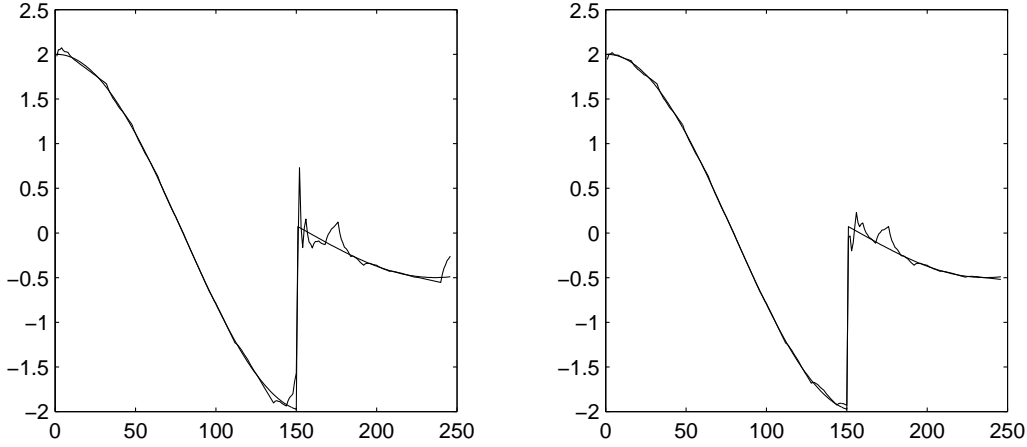


Figure 3.6: On the left is approximation by the standard wavelet approach and on the right with the modified approach. Same number of coefficients (22) are used in both cases.

Therefore, the modification of all the big jumps is not necessary and it may turn out to be an extra computation. For this reason, we introduce a second threshold  $T_2$  in our algorithm and its value depends upon the number  $N$ . We have used  $T_2 = \frac{T_1}{3}$ ,  $T_2 = \frac{T_1}{2}$  and  $T_2 = T_1$  in our experiments.

That is, we have modified only those jumps whose magnitude is greater than the standard deviation of the signal and generate a wavelet coefficients of significant magnitude depending upon  $N$ . This approach has worked well. There may be room to find a better procedure for finding the threshold especially  $T_2$  that may improve the performance of our algorithm.

Figure 3.6 shows the approximation of the piecewise smooth signal of figure 3.4 with the standard approach and with the new approach. Original signal has 248 data points, while in the approximation only 22 coefficients are used in both cases. One can see the reduction in the Gibbs Phenomenon in the approximation by the new approach compared to the standard wavelet approach.



Figure 3.7: On the left is approximation by the standard wavelet approach and on the right with the modified approach. Same number of coefficients (2048) are used in both cases.

We use many standard images (usually used by the image processing people), to compare the improvement in the quality of approximation by the new approach. We note that the visual quality of the approximated images by the new approach is better compared to the approximated images with the standard wavelet approach while using the same number of coefficients. From figures 3.7, 3.9, 3.11, 3.13, 3.15, 3.16 and 3.17, one can see the reduction in the Ringing Effect in the images approximated by our approach compared to the standard wavelet approach.

From our experiments, we have found that our approach gives better results even if we use small number of the coefficients compared to the standard wavelet approach. In figures 3.8, 3.10, 3.12 and 3.14, we give comparison of our approach with the standard wavelet approach. Only 66% of the coefficients used in the standard wavelet approach are used in our approach and we have almost same or better quality of approximated images.



Figure 3.8: On the left is approximation by the standard wavelet approach with 2048 largest coefficients and on the right with the modified approach with 1352 largest coefficients (1352 is 66% of 2048).

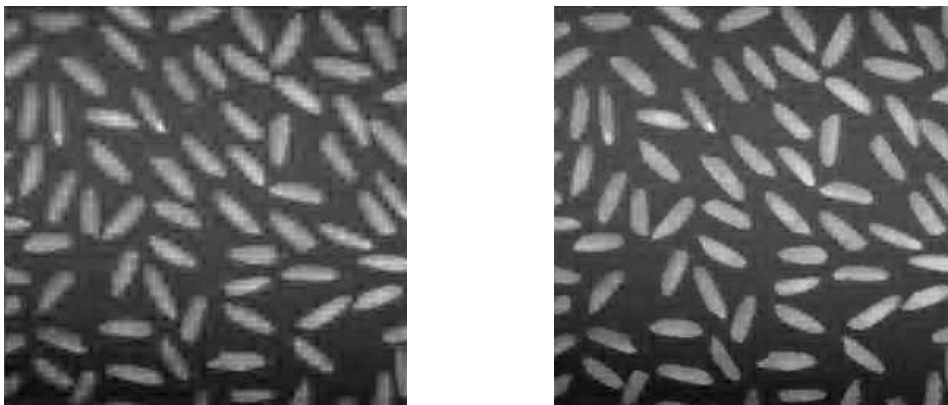


Figure 3.9: On the left is approximation by the standard wavelet approach and on the right with the modified approach. Same number of coefficients (2048) are used in both cases. The gain in PSNR is 3.6216.

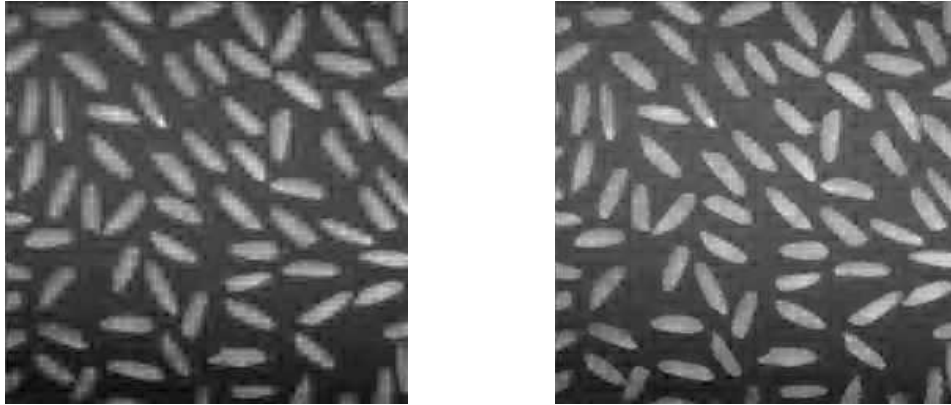


Figure 3.10: On the left is approximation by the standard wavelet approach with 2048 largest coefficients and on the right with the modified approach with 1352 largest coefficients (1352 is 66% of 2048).

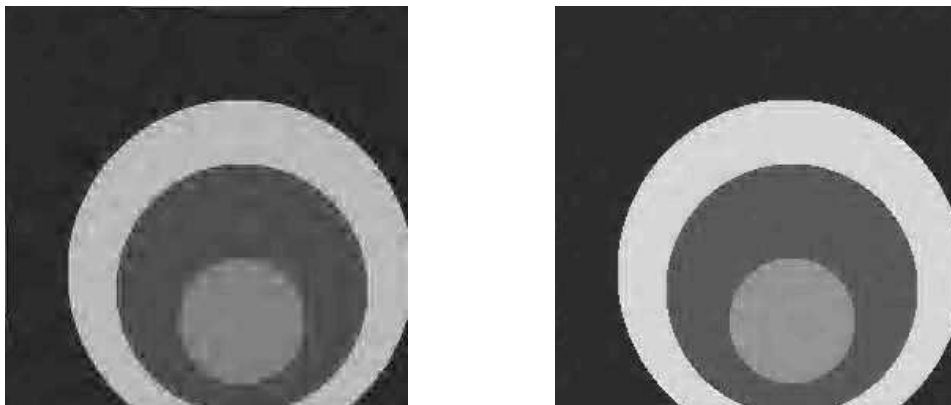


Figure 3.11: On the left is approximation by the standard wavelet approach and on the right with the modified approach. Same number of coefficients (1024) are used in both cases. The gain in PSNR is 7.2727.

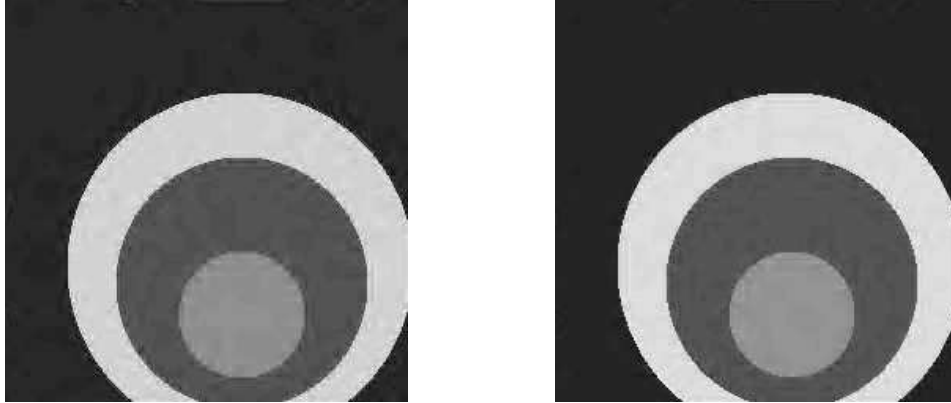


Figure 3.12: On the left is approximation by the standard wavelet approach with 2048 largest coefficients and on the right with the modified approach with 1352 largest coefficients (1352 is 66% of 2048).



Figure 3.13: On the left is approximation by the standard wavelet approach and on the right with the modified approach. Same number of coefficients (4096) are used in both cases. The gain in PSNR is 1.6989.



Figure 3.14: On the left is approximation by the standard wavelet approach with 6000 largest coefficients and on the right with the modified approach with 3960 largest coefficients (3960 is 66% of 6000).



Figure 3.15: On the left is approximation by the standard wavelet approach and on the right with the modified approach. Same number of coefficients (4096) are used in both cases. The gain in PSNR is 1.8654.



Figure 3.16: On the left is approximation by the standard wavelet approach and on the right with the modified approach. Same number of coefficients (4096) are used in both cases. The gain in PSNR is 1.1147.



Figure 3.17: On the left is approximation by the standard wavelet approach and on the right with the modified approach. Same number of coefficients (2048) are used in both cases. The gain in PSNR is 1.8526.

### 3.3 Optimizing our Transform

As we have noted in the previous section the parameter  $a$  in the smoothing transform

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ -a & 1 & a \\ 0 & 0 & 1 \end{pmatrix}$$

depends upon the wavelet filter used and the location of the singularity in the discrete signal. We can find the value of this parameter  $a$  which makes the transform  $\mathbf{A}$  optimal in the sense that our algorithm can perform better. In this section we compute the value of  $a$  for the Daubechies wavelet filters of length 4 and 6. To find this parameter for other wavelet filters, the procedure is similar.

#### 3.3.1 Special Case for Daubechies wavelet filter of length 4

For the Daubechies wavelet filter of length 4 (see [11]), we have,

- Low pass filter  $h = [\frac{1+\sqrt{3}}{4\sqrt{2}}, \frac{3+\sqrt{3}}{4\sqrt{2}}, \frac{3-\sqrt{3}}{4\sqrt{2}}, \frac{1-\sqrt{3}}{4\sqrt{2}}]$ ;
- High pass filter  $g = [\frac{1-\sqrt{3}}{4\sqrt{2}}, \frac{-3+\sqrt{3}}{4\sqrt{2}}, \frac{3+\sqrt{3}}{4\sqrt{2}}, \frac{-1-\sqrt{3}}{4\sqrt{2}}]$ ;
- For simplicity we assume here a piecewise constant signal  $\mathbf{f} = \{f_i\}$ . If there is a jump at the  $i^{th}$  step then without loss of generality, we assume that

$$f_{i-2} = p, f_{i-1} = p, f_i = p$$

and

$$f_{i+1} = q, f_{i+2} = q, f_{i+3} = q$$

for some constants  $p$  and  $q$ . Moreover, we assume that the jumps occur at least 3 pixels apart.



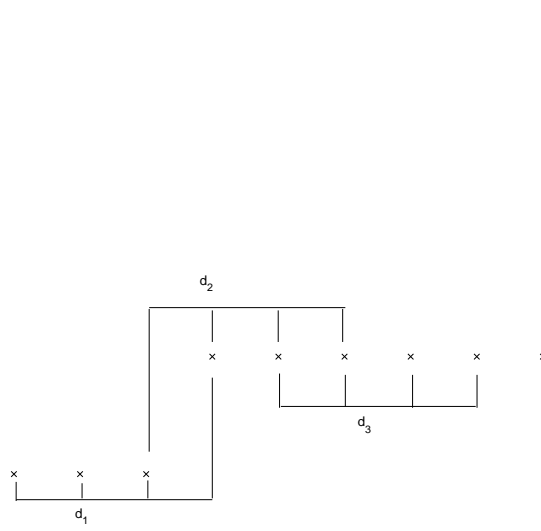


Figure 3.18: Implementation of Daubechies filter of length 4 on a signal with the discontinuity at odd location

### Odd location

When the discontinuity is at odd location (i.e.  $i$  is odd ), we may have only 2 wavelet coefficients of significant magnitude. Let  $d_1$  and  $d_2$  be these wavelet coefficients then

$$d_1 = g_1 f_{i-2} + g_2 f_{i-1} + g_3 f_i + g_4 f_{i+1} = (g_1 + g_2 + g_3)p + g_4 q = g_4(q - p)$$

$$d_2 = g_1 f_i + g_2 f_{i+1} + g_3 f_{i+2} + g_4 f_{i+3} = g_1 p + (g_2 + g_3 + g_4)q = -g_1(q - p)$$

Let  $\tilde{d}_1$  and  $\tilde{d}_2$  be the corresponding wavelet coefficients for the modified signal (here the value of  $f_i$  is modified), Then

$$\tilde{d}_1 = d_1 + a g_3 (q - p) = (g_4 + a g_3)(q - p)$$

$$\tilde{d}_2 = d_2 + a g_1 (q - p) = -(1 - a)g_1(q - p)$$

We need to choose  $a$  for which both  $\tilde{d}_1$  and  $\tilde{d}_2$  are less significant.

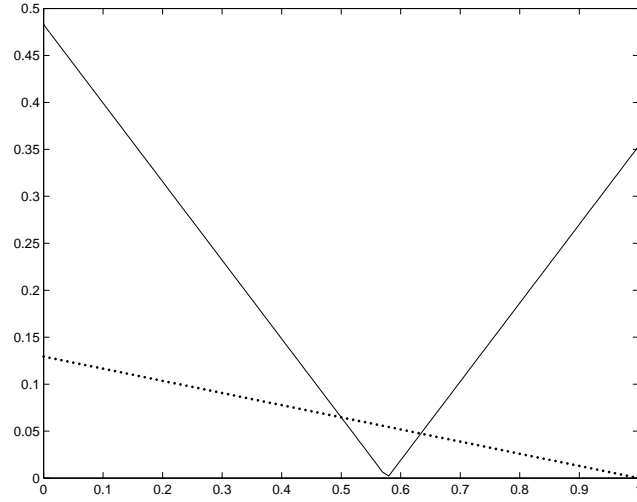


Figure 3.19: Graph of  $|\tilde{d}_1|$  (solid line) and  $|\tilde{d}_2|$  (dotted line) for odd location and the Daubechies wavelet filter of length 4. Here we assumed  $|q - p| = 1$

Visual inspection of the graph of  $|\tilde{d}_1|$  and  $|\tilde{d}_2|$  given in the figure 3.19 suggests that  $0.50 \leq a \leq 0.64$  will give better performance. However, we can find the value of the parameter  $a$ , by minimizing  $|\tilde{d}_1|$  and  $|\tilde{d}_2|$  in the sense of different norms. The parameter value in the sense of  $\ell_1$  norm is  $a = .5777 \dots$ , in the sense of energy norm (i.e  $\ell_2$  norm) is  $a = .58722 \dots$ , and in the sense of  $BV$  norm is  $a = \frac{1}{2}$ .

### Even Location

When the discontinuity is in an even location (i.e.  $i$  is even) then we may have only one significant wavelet coefficient say  $d_2$  then

$$d_2 = (g_3 + g_4)(q - p)$$

and the neighboring wavelet coefficients are zero say  $d_1 = 0$  and  $d_2 = 0$ .

Let  $\tilde{d}_1$ ,  $\tilde{d}_2$  and  $\tilde{d}_3$  are the corresponding wavelet coefficients for the modified signal. In this case the modification of the signal is not as simple. If we modify  $f_i$  then  $\tilde{d}_1$  will

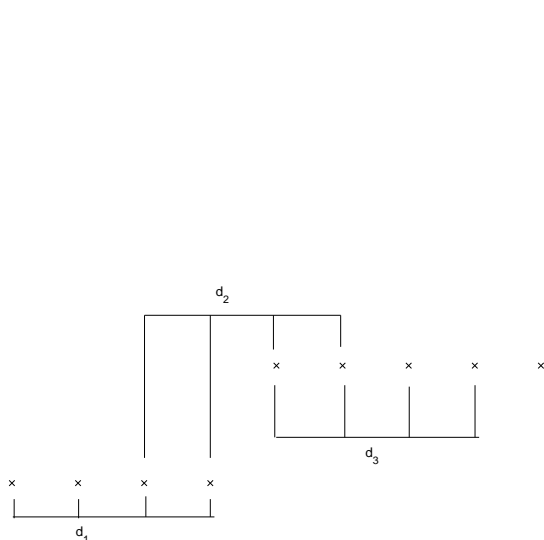


Figure 3.20: Implementation of Daubechies filter of length 4 on a signal with the discontinuity at even location

grow and if we modify  $f_{i+1}$  then  $\tilde{d}_3$  will grow. But the growth in  $\tilde{d}_3$  turns out to be small compared to the growth in  $\tilde{d}_1$ . Therefore, we decide to modify  $f_{i+1}$ , and the corresponding values of the wavelet coefficients are

$$\tilde{d}_2 = (g_4 + (1 - a)g_3)(q - p)$$

$$\tilde{d}_3 = -ag_1(q - p).$$

Since  $\tilde{d}_1$  is not effected by this modification, we still have  $\tilde{d}_1 = 0$ .

The graph of  $|\tilde{d}_2|$  and  $|\tilde{d}_3|$  shown in the figure 3.21 suggests that  $|\tilde{d}_2|$  and  $|\tilde{d}_3|$  will be less significant for  $0.36 \leq a \leq 0.50$ . The graph of modified wavelet coefficients for odd location shown in the figure 3.21 is the mirror image of the graph of modified wavelet coefficients for even location shown in the figure 3.19. Therefore, the value of the parameter  $a$ , which minimizes the modified wavelet coefficients for even location is just the  $1 - a$  value of the odd location.

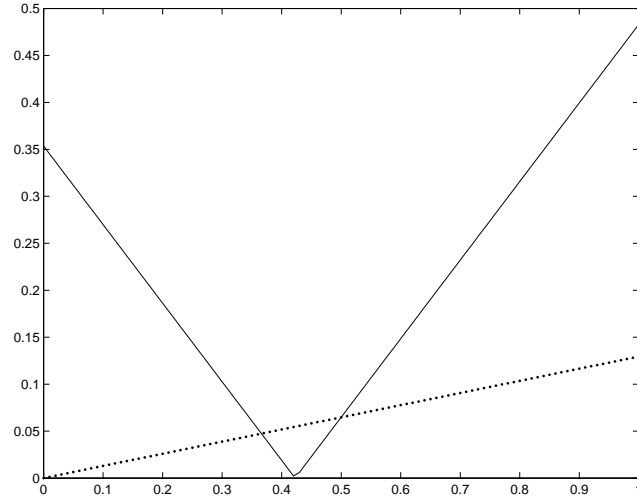


Figure 3.21: Graph of  $|\tilde{d}_2|$  (solid line) and  $|\tilde{d}_3|$  (dotted line) for even location and the Daubechies wavelet filter of length 4. Here we assumed  $|q - p| = 1$

For the even location, the parameter value in the sense of  $\ell_1$  norm is  $a = .42222 \dots$ , in the sense of energy norm (i.e  $\ell_2$  norm) is  $a = .41277 \dots$ , and in the sense of  $BV$  norm is  $a = \frac{1}{2}$ .

### 3.3.2 Special Case for Daubechies wavelet filter of length 6

In the case of Daubechies wavelet filter of length 6 (see [11]), we have

- Low pass filter  $h = [0.3327, 0.8069, 0.4599, -0.1350, -0.0854, 0.0352]$ ;
- High pass filter  $g = [0.0352, 0.0854, -0.1350, -0.4599, 0.8069, -0.3327]$ .

Here, we assume that we have a piecewise constant signal (as in the above case) with isolated singularities at least 5 pixels apart.

#### Odd Location

If the singularity lies in an odd location (i.e.  $i = \text{odd}$ ), then there may be only 3 significant wavelet coefficients corresponding to this singularity. Let  $d_1$ ,  $d_2$  and  $d_3$  be these coefficients. Then we have

$$d_1 = g_1p + g_2p + g_3p + g_4p + g_5p + g_6q = g_6(q - p)$$

$$d_2 = g_1p + g_2p + g_3p + g_4q + g_5q + g_6q = (g_4 + g_5 + g_6)(q - p)$$

$$d_3 = g_1p + g_2q + g_3q + g_4q + g_5q + g_6q = -g_1(q - p)$$

Let  $\tilde{d}_1$ ,  $\tilde{d}_2$  and  $\tilde{d}_3$  be the corresponding wavelet coefficients for the modified signal (i.e. modifying  $f_i$ ). Then

$$\tilde{d}_1 = d_1 + ag_5(q - p)$$

$$\tilde{d}_2 = d_2 + ag_3(q - p)$$

$$\tilde{d}_3 = d_3 + ag_1(q - p)$$

Figure 3.22 shows the graph of  $\tilde{d}_1$ ,  $\tilde{d}_2$  and  $\tilde{d}_3$ , which suggests that  $\tilde{d}_1$ ,  $\tilde{d}_2$  and  $\tilde{d}_3$  are less significant for  $0.35 \leq a \leq 0.45$ . However, we can compute the value of  $a$  for which the modified wavelet coefficients  $\tilde{d}_1$ ,  $\tilde{d}_2$  and  $\tilde{d}_3$  are minimized in the sense of different norms. Thus the parameter value for the modification of odd location in the sense of  $\ell_1$  norm is  $a = .4123 \dots$ , in the sense of energy norm (i.e  $\ell_2$  norm) is  $a = .405077 \dots$ , and in the sense of  $BV$  norm is  $a = .3684 \dots$ .

### Even Location

Let assume that the singularity lies in the even location (i.e.  $i = \text{even}$ ). There may be only 2 significant wavelet coefficients corresponding to this singularity. Let  $d_2$  and  $d_3$  be these coefficients. Then we have

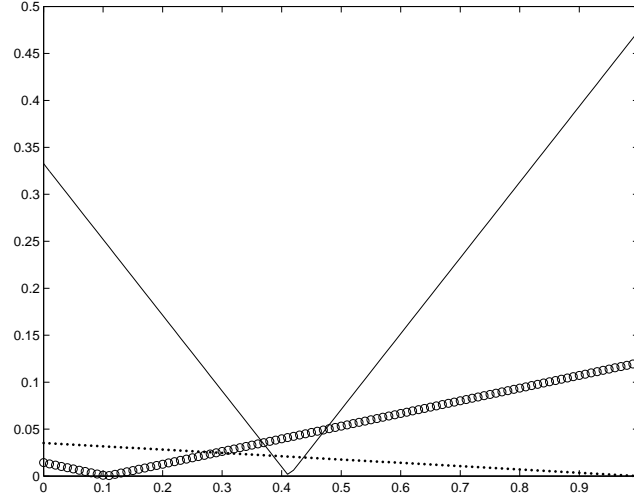


Figure 3.22: Graph of  $|\tilde{d}_1|$  (solid line),  $|\tilde{d}_2|$  (thick dotted line) and  $|\tilde{d}_3|$  (dotted line) for odd location and the Daubechies wavelet filter of length 6. Here we assumed  $|q - p| = 1$

$$d_2 = g_1p + g_2p + g_3p + g_4p + g_5q + g_6q = (g_5 + g_6)(q - p)$$

$$d_3 = g_1p + g_2p + g_3q + g_4q + g_5q + g_6q = -(g_1 + g_2)(q - p)$$

Now in the modification process , if we modify  $f_i$  or  $f_{i+1}$ , another coefficient may grow to significant magnitude. But the magnitude of this newly enlarged coefficient is smaller in case of the modification of  $f_{i+1}$  compared to  $f_i$ . Therefore, we choose to modify  $f_{i+1}$ . As a result, we have,

$$\tilde{d}_1 = 0$$

$$\tilde{d}_2 = d_2 - ag_5(q - p)$$

$$\tilde{d}_3 = d_3 - ag_3(q - p)$$

$$\tilde{d}_4 = -ag_1(q - p)$$

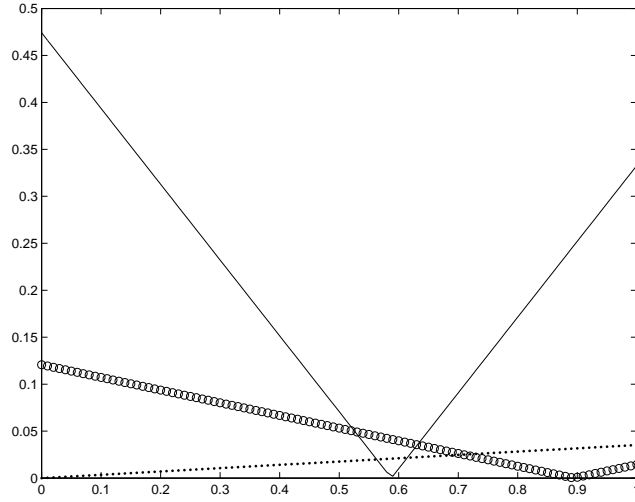


Figure 3.23: Graph of  $|\tilde{d}_2|$  (solid line),  $|\tilde{d}_3|$  (thick dotted line) and  $|\tilde{d}_4|$  (dotted line) for even location and the Daubechies wavelet filter of length 6. Here we assumed  $|q - p| = 1$

The graphs of the modified wavelet coefficients  $\tilde{d}_2$ ,  $\tilde{d}_3$  and  $\tilde{d}_4$  are shown in the figure 3.23, which suggest that  $\tilde{d}_2$ ,  $\tilde{d}_3$  and  $\tilde{d}_4$  will be less significant for  $0.55 \leq a \leq 0.65$ . The graph of the modified wavelet coefficients for odd location given in figure 3.22 is the mirror image of the graph of the modified wavelet coefficients at even location given in figure 3.23. Therefore, the value of the parameter  $a$ , which minimizes the modified wavelet coefficients for even location is just the  $1 - a$  value of the parameter value that minimizes the modified wavelet coefficients at odd location.

We can compute the value of  $a$  for which the modified wavelet coefficients  $\tilde{d}_2$ ,  $\tilde{d}_3$  and  $\tilde{d}_4$  are minimized in the sense of different norms. Thus the parameter value for the modification of even location in the sense of  $\ell_1$  norm is  $a = .5876 \dots$ , in the sense of energy norm (i.e  $\ell_2$  norm) is  $a = .594922 \dots$ , and in the sense of  $BV$  norm is  $a = .6315 \dots$ .

### 3.4 Smoothing Transform of Higher Order

Let  $f$  be a 1-D signal with a big jump in its values  $f_{i-1}$  and  $f_i$ . In order to make the transition smooth, we can modify more than one point of the signal in the neighborhood of the singularity. Previously, we used  $3 \times 3$  transform with single parameter  $a$  to modify a single point in the singularity. If we want to modify 2 points (i.e.  $f_{i-1}$  and  $f_i$ ) then we would need a  $4 \times 4$  transform with 2 parameters and, similarly, to modify 3 points, we would need a  $5 \times 5$  transform with more than three parameters.

Here is given an example of  $5 \times 5$  transform and the modification procedure.

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -a & 1 & a-b & b & 0 \\ e & c-e & 1 & d-c & -d \\ 0 & h & g-h & 1 & -g \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

where  $a, b, c, d, e, g$  and  $h$  are some non negative constants depending upon the wavelet filter used and the location of the singularity in the signal.

When this transform is applied to the segment  $(f_{i-2}, f_{i-1}, f_i, f_{i+1}, f_{i+2})$  of the signal, then we obtain the new signal values

$$\begin{aligned} \tilde{f}_{i-2} &= f_{i-2} \\ \tilde{f}_{i-1} &= f_{i-1} + a(f_i - f_{i-2}) + b(f_{i+1} - f_i) \\ \tilde{f}_i &= f_i - c(f_{i+1} - f_{i-1}) - d(f_{i+2} - f_{i+1}) - e(f_{i-1} - f_{i-2}) \\ \tilde{f}_{i+1} &= f_{i+1} - g(f_{i+2} - f_i) - h(f_i - f_{i-1}) \\ \tilde{f}_{i+2} &= f_{i+2} \end{aligned}$$

where  $\tilde{f}_i$  represents the modified value of  $f_i$ . The above system of equations can be



expressed in a matrix form

$$\tilde{\mathbf{f}} = \mathbf{A}\mathbf{f}$$

where

$$\tilde{\mathbf{f}} = \begin{pmatrix} \tilde{f}_{i-2} \\ \tilde{f}_{i-1} \\ \tilde{f}_i \\ \tilde{f}_{i+1} \\ \tilde{f}_{i+2} \end{pmatrix}$$

and

$$\mathbf{f} = \begin{pmatrix} f_{i-2} \\ f_{i-1} \\ f_i \\ f_{i+1} \\ f_{i+2} \end{pmatrix}$$

There may be many choices in selecting the values of the parameters  $a, b, c, d, e, g$  and  $h$ . In order to reduce the number of these parameters, we can take  $h = b, g = a$  and  $e = d$ . Then the matrix  $\mathbf{A}$  will look like

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -a & 1 & a-b & b & 0 \\ d & c-d & 1 & -(c-d) & -d \\ 0 & b & a-b & 1 & -a \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Experiments show that increasing the order of the transform does not increase the performance of the algorithm. Sometimes it is even worst for 2-D signals. Since we apply the algorithm along each column and each row (i.e. 1-D implementation) while processing a column we don't know about the jumps in other dimension. Thus modifying more than

one point may be dangerous. Therefore, it does not seem beneficial to use a higher order transform. To keep the computation simple and at the same time to get the best output, we use  $3 \times 3$  transform for orthogonal wavelets.

### 3.5 Approximation Error for Piecewise Smooth Function

Let  $f(x)$  be a piecewise smooth function with a discontinuity in its function values at the point  $x_0$ . Then we can modify this signal in the sense of our algorithm (for fixed  $\Delta(x)$ ) as

$$\tilde{f}(x) = \begin{cases} f(x) + ag(x) & \text{if } x \in (x_0 - \Delta(x), x_0] \\ f(x) - (1 - a)g(x) & \text{if } x \in (x_0, x_0 + \Delta(x)) \\ f(x) & \text{otherwise} \end{cases}$$

where

$$g(x) = \left[ \frac{(x - x_{-1})(x_1 - x)}{(\Delta(x))^2} \right]^2 (J_f + J_{f'}(x - x_0)),$$

$$J_f = f(x_0^+) - f(x_0^-),$$

$$J_{f'} = f'(x_0^+) - f'(x_0^-),$$

$x_1 = x_0 + \Delta(x)$ ,  $x_{-1} = x_0 - \Delta(x)$  and  $a \in (0, 1)$ .

Thus we have  $g(x_0) = J_f$  and therefore,

$$\tilde{f}(x_0^-) = f(x_0^-) + aJ_f$$

and

$$\tilde{f}(x_0^+) = f(x_0^+) - (1 - a)J_f$$

Thus

$$\tilde{f}(x_0^+) - \tilde{f}(x_0^-) = 0.$$

Similarly,

$$g'(x) = 2\left[\frac{(x-x_{-1})(x_1-x)}{(\Delta(x))^2}\right]\left(\frac{(x_1-x)-(x-x_{-1})}{(\Delta(x))^2}\right)(J_f+J_{f'}(x-x_0))+\left[\frac{(x-x_{-1})(x_1-x)}{(\Delta(x))^2}\right]^2 J_{f'}$$

Which gives  $g'(x_0) = J_{f'}$  and consequently  $\tilde{f}'(x_0^+) = \tilde{f}'(x_0^-)$ .

Thus with such a modification, the modified data agrees with a function of increased smoothness at the point of discontinuity while keeping the same smoothness at the other points. If we denote the smoothing transform by  $T$ , then we can write

$$\tilde{f} = Tf.$$

If  $\|\tilde{f}'(x)\|$  is bounded then by Theorem 2.2.1,

$$\|\tilde{f}(x) - (\tilde{f})_j(x)\| \leq C\Delta(x)\|\tilde{f}'(x)\|$$

where  $(\tilde{f})_j(x)$  is the  $j^{\text{th}}$  level wavelet approximation of  $\tilde{f}(x)$ .

This result holds for  $L^2$  norm in general. Moreover, if the scaling function and its wavelet have finite support, then it also holds for the  $L^\infty$  norm.

In the sense of our algorithm,

$$\begin{aligned} \|f(x) - T^{-1}(\tilde{f})_j(x)\| &= \|T^{-1}Tf(x) - T^{-1}(\tilde{f})_j(x)\| \\ &\leq \|T^{-1}\| \|\tilde{f}(x) - (\tilde{f})_j(x)\| \\ &\leq \|T^{-1}\| C\Delta(x)\|\tilde{f}'(x)\| \end{aligned}$$

Here  $\|T\|^2$  and  $\|T^{-1}\|^2$  are finite values (see Appendix for details).

In some sense we have an error bound for our approximation to the piecewise smooth function.

## 3.6 Numerical Results

In our experiments, we found that the new algorithm works better than standard wavelet method on a wide range of commonly used images. It is interesting to note that in some figures, we used only 66% of the largest wavelet coefficients in new method than largest wavelet coefficients used in the standard wavelet approximation and still get visually better result. Actually the edges looks more well formed and sharper than with the wavelet approximation.

In our numerical computation, we performed the smoothing procedure for the first three levels of wavelet decomposition and reconstruction. For the remaining level it is standard wavelet decomposition and reconstruction.

We are presenting the results for Daubechies wavelet filter of length 4. For other orthogonal/Bi-orthogonal wavelet filters, the performance of our algorithm is similar.

### 3.6.1 Daubechies Wavelet Filter of length 4 is used for the following results:

In all of our 2-D experiments with Daubechies wavelet filter of length 4, we used the transform  $\mathbf{A}$  with  $a = .58$ , for odd location of the singularities and  $a = .42$  for the even location of the singularities. Figure 3.24 shows the graphic representation of the improvement in PSNR ( i.e. Peak Signal to Noise Ratio) value by increasing the modified points for the rice image .

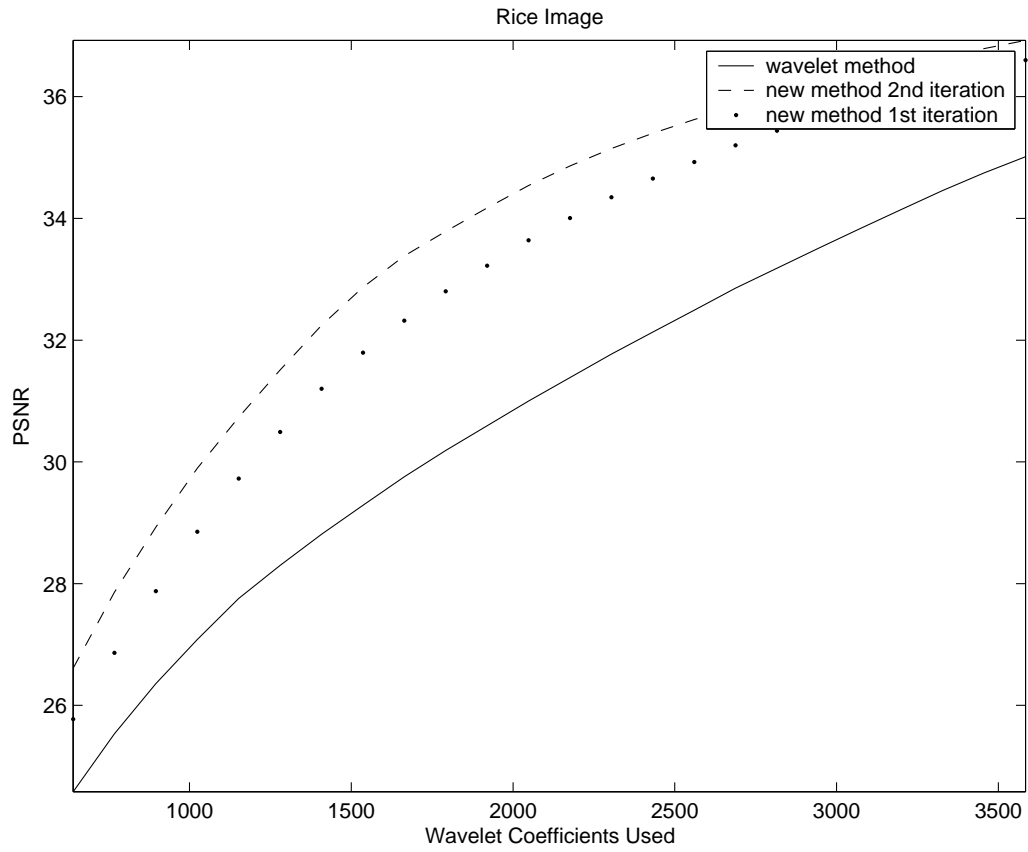


Figure 3.24: Comparison in PSNR value for the rice image (one can see how the modification effects the PSNR values): first iteration means signal is modified only in 1st level of decomposition and 2nd iteration means that signal is modified during 1st and 2nd level of decomposition.

Table 3.1: **Circles Image** of size 256 by 256. Here the signal is modified at 2013 points.

Coefficients Used	PSNR using New approach	PSNR using Standard wavelet	Gain in PSNR value
1000	32.4509	25.7956	6.6554
2000	38.4554	30.7406	7.7148
3000	43.4929	35.4181	8.0748
4000	49.4361	40.4237	9.0124
5000	55.6852	45.7293	9.9558

Table 3.2: **Rice Image** of size 256 by 256. Here the signal is modified at 3529 points.

Coefficients Used	PSNR using New approach	PSNR using Standard wavelet	Gain in PSNR value
1000	29.1331	24.8227	4.3104
2000	31.5276	28.2634	3.2642
3000	33.0235	30.6907	2.3328
4000	34.3167	32.6848	1.6319
5000	35.4668	34.2948	1.1720
6000	36.5189	35.6646	0.8543

Table 3.3: **House Image** of size 256 by 256. Here the signal is modified at 2245 points.

Coefficients Used	PSNR using New approach	PSNR using Standard wavelet	Gain in PSNR value
1000	30.3072	27.9257	2.3815
2000	32.6140	30.9710	1.6430
3000	34.0061	33.0017	1.0044
4000	35.1316	34.5368	0.5948

Table 3.4: **Camerman Image** of size 256 by 256. Here the signal is improved at 3881 places.

Coefficients Used	PSNR using New approach	PSNR using Standard wavelet	Gain in PSNR value
1000	26.3651	23.8168	2.5483
2000	28.4076	26.2352	2.1724
3000	29.7057	27.9664	1.7393
4000	30.7931	29.4136	1.3795
5000	31.7983	30.6863	1.1120
6000	32.7679	31.8399	0.9280

Table 3.5: **Peppers Image** of size 512 by 512. Here the signal is modified at 5749 points.

Coefficients Used	PSNR using New approach	PSNR using Standard wavelet	Gain in PSNR value
3000	29.4350	27.4841	1.9509
4000	30.4385	28.6956	1.7429
5000	31.1344	29.6433	1.4911
6000	31.6998	30.4215	1.2783
7000	32.1624	31.0770	1.0819
8000	32.5654	31.6489	0.9518

Table 3.6: **Lena Image** of size 512 by 512. Here the signal is modified at 7647 points.

Coefficients Used	PSNR using New approach	PSNR using Standard wavelet	Gain in PSNR value
3000	29.4746	27.7135	1.7611
4000	30.3250	28.7338	1.5912
5000	30.9729	29.5729	1.4000
6000	31.5184	30.2887	1.2307
7000	32.0007	30.9188	1.0819
8000	32.4324	31.4806	0.9518



Table 3.7: **Barbara Image** of size 512 by 512. Here the signal is improved at 22034 places.

Coefficients Used	PSNR using New approach	PSNR using Standard wavelet	Gain in PSNR value
3000	24.2032	23.1689	1.0334
4096	24.7961	23.7732	1.0229
5000	25.2124	24.2177	0.9947
6000	25.6253	24.6718	0.9535
7000	25.9943	25.0916	0.9027
8000	26.3381	25.4867	0.8514

### 3.6.2 Comparison: New Approach with Daubechies Wavelet Filter of length 4 with standard Bi-orthogonal Wavelet Filter of length 9

Since a Bi-orthogonal wavelet filter of length 9 is smoother and has greater vanishing moments than the Daubechies orthogonal wavelet of vanishing moment 2 (corresponding filter of length 4), better results are expected while using Bi-orthogonal filter as compared to Daubechies filter of length 4. But we have noted that by our approach, Daubechies filter of length 4 can beat the Bi-orthogonal filter when a small number of coefficients in approximating the signal/image is used.

We have observed that our algorithm gave more gain in terms of Peak Signal to Noise Ratio (PSNR) for less smooth filters. One can see that the gain in PSNR by Daubechies filter of length 4 is more than the gain using Bi-orthogonal filter for the small number of coefficients. However if we increase the number of coefficients then Bi-orthogonal starts giving better performance.

Table 3.8: **Cameraman Image:** For Daubechies wavelet filter signal modified at 3881 points and for Bi-orthogonal filter at 3655 points, we find

Coefficients Used	PSNR by New approach Daubechies filter length 4	PSNR using Bi-Orthogonal Filter	PSNR by New approach Bi-orthogonal Filter
1000	26.3651	24.6009	25.7175
2000	28.4076	27.1719	28.0827
3000	29.7057	28.9593	29.6841
4000	30.7931	30.3710	30.9480
5000	31.7983	31.5813	32.0891

Table 3.9: **Rice Image:** For Daubechies wavelet filter signal modified at 3529 points and for Bi-orthogonal filter at 3294 points, we find

Coefficients Used	PSNR by New approach Daubechies filter length 4	PSNR using Bi-Orthogonal Filter	PSNR by New approach Bi-orthogonal Filter
1000	29.1331	26.9447	28.4236
2000	31.5276	30.8519	32.3791
3000	33.0235	33.6531	35.2594
4000	34.3167	35.8568	37.0247
5000	35.4668	37.5728	38.4827

Table 3.10: **Peppers Image:** For Daubechies wavelet filter signal modified at 5749 points and for Bi-orthogonal filter at 5550 points, we find

Coefficients Used	PSNR by New approach Daubechies filter length 4	PSNR using Bi-Orthogonal Filter	PSNR by New approach Bi-Orthogonal Filter
3000	29.4350	28.5693	29.5872
4000	30.4385	29.8234	30.7300
5000	31.1344	30.8095	31.5566
6000	31.6998	31.6044	32.2201
7000	32.1624	32.2457	32.7575

## Chapter 4

# Radon Transform and its Applications

There are several approaches to find edges or discontinuities in the given image. Most common approaches for this purpose are the use of certain masks, first or second derivatives, Graph-Theoretic method and Radon Transform etc. as described in [1], [10], [12], [18] and [20]. In this chapter, we have used Radon Transform to find the linear edges present in the given image. First, we give a new approach to discretize the Radon Transform and then we apply it to digital images to find edges. We also give some applications of Radon Transform in image approximation. Some numerical results are also presented.

### 4.1 Radon Transform

Given an integrable function  $f(x, y)$ , Radon Transform  $RA_f$  is defined by

$$RA_f(\theta, t) = \int_R \int_R f(x, y) \delta(x \cos \theta + y \sin \theta - t) dx dy. \quad (4.1)$$

Since the Radon Transform is radial in nature, its implementation to the digital images is not straight-forward. A simple discretization of the continuous formula will not work. It requires interpolation and thus results in transform which is either redundant or cannot be perfectly reconstructed.

Here is a brief review of some of the implementations of Radon Transform and the problems associated with these approaches.

#### 4.1.1 Fourier Approach to Compute Radon Transform

The Radon Transform has a very interesting property called Projection-Slice Theorem [12], [14] and [15],

$$\hat{f}(\lambda \cos \theta, \lambda \sin \theta) = \int_R RA_f(\theta, t) e^{-i\lambda t} dt.$$

This means that for the given signal  $f(x, y)$ , find its 2-D Fourier Transform  $\hat{f}(\lambda \cos \theta, \lambda \sin \theta)$  and then the Radon Transform  $RA_f$  is simply computed by taking the 1-D Inverse Fourier Transform of  $\hat{f}(\lambda \cos \theta, \lambda \sin \theta)$  along the radial line passing through the origin.

In practice, one has to take the following three steps to compute the Radon Transform of the given digital image  $[f(i, j)]$ ,  $0 \leq i, j < n - 1$  ;

1. **2-D FFT** (Fast Fourier Transform): Compute the 2-D FFT of  $f$  i.e.  $[\hat{f}(w_1, w_2)]$ ,  $-\frac{n}{2} \leq w_1, w_2 \leq \frac{n}{2} - 1$ .
2. **Change from Cartesian to Polar**: Substitute the sampled values of the Fourier Transform in the cartesian grid with the sampled values of  $\hat{f}$  in polar grid (the lattice where the points fall on the lines going through the origin) by using some interpolation scheme or any other method.

3. **1-D IFFT** (Inverse Fast Fourier Transform): Compute the 1-D IFFT on each line through the origin.

This scheme is widely used in the medical imaging and synthetic aperture radar imaging as discussed in [12]. Such schemes (involving interpolation) are either redundant or have greater approximation errors as pointed out in [12] and [15].

#### 4.1.2 Finite Radon Transform

The Finite Radon Transform (FRAT) as described in [8], [14] and [15], is the summation of image pixels over a certain set of lines, which are defined in some finite geometry in a similar way as lines for continuous Radon Transform in the Euclidean geometry. Let  $Z_p = \{0, 1, 2, \dots, p-1\}$  be a finite field, where  $p$  is a prime number.

The FRAT of a real function  $f(x, y)$  on the finite grid  $Z_p^2$  is defined by

$$FRAT_f(k, l) = \frac{1}{\sqrt{p}} \sum_{(i,j) \in L_{k,l}} f[i, j]. \quad (4.2)$$

Here  $FRAT_f(k, l)$  is called the Radon coefficient along the line  $L_{k,l}$  and the line  $L_{k,l}$  denotes the set of points that make up a line on the lattice  $Z_p^2$ ,

$$L_{k,l} = \begin{cases} \{(i, j) : j = (ki + l) \bmod(p), i \in Z_p\} & \text{if } 0 \leq k < p \\ \{(l, j) : j \in Z_p\} & \text{if } k = p. \end{cases} \quad (4.3)$$

The inverse transform is obtained through the Finite Back-Projection Operator defined in [14] and [15] as a sum of Radon coefficients of all the lines that go through the given point.

That is

$$FBP_f(i, j) = \frac{1}{\sqrt{p}} \sum_{(k,l) \in P_{i,j}} FRAT_f(k, l), \quad (4.4)$$

where  $P_{i,j}$  denote the set of indices of all the lines that go through the point  $(i, j) \in Z_p^2$ .

There are two problems associated with this discrete implementation as mentioned in [14] and [15], one is that  $p$  in the size of the block, is a prime number but in practice we do not always have blocks of prime number length. The second is the wrap-around effect (line aliasing) of this finite implementation.

### 4.1.3 Our Approach to Distcretize Radon Transform

In order to avoid the problems associated with the implementation of Radon Transform as mentioned in the previous sections, we introduce a different scheme. We express the line  $x\cos\theta + y\sin\theta = t$  in the Radon Transform by slope-point form  $y - y' = m'(x - x')$  i.e. the line with slope  $m' = \tan\theta$  and through the point  $p'$  having coordinates  $(x', y')$  in the spatial domain corresponding to the point  $(p', \theta)$  in the Radon space. Here by the Radon space, we mean the space of all order pairs  $(p, \theta)$  with  $p$  is a point in XY-plane and  $0 \leq \theta \leq \pi$ . So we have the Radon Transform

$$RA_f(p', \theta) = \int_R \int_R f(x, y) \delta(y - y' - (x - x') \tan\theta) dx dy. \quad (4.5)$$

Equivalently, we can write the above equation as

$$RA_f(p', m') = \int_R \int_R f(x, y) \delta(y - y' - (x - x') m') dx dy. \quad (4.6)$$

It can be discretized for data on a  $m \times n$  grid. To keep the implementation simple and easy, we discretize slope with equal step size instead of discretizing  $\theta$ .

$$RA_f(p'_{kl}, m') = \sum_{i=1}^m \sum_{j=1}^n f(i, j) \delta(j - m'i - (l - m'k)), \quad (4.7)$$

where  $p'_{kl}$  represents a point  $(k, l)$  of grid, with  $1 \leq k \leq m$ ,  $1 \leq l \leq n$ .

Equation 4.7 can be expressed as

$$RA_f(p'_{kl}, m') = \sum_{(i,j) \in L_{(p'_{kl}, m')}} f(i, j), \quad (4.8)$$

where  $L_{(p'_{kl}, m')}$  represents a line through a point  $p'_{kl}$  having slope  $m'$  and  $RA_f(p'_{kl}, m')$  is the corresponding Radon Coefficient.

To make a fair comparison among Radon coefficients, one should normalize equation 4.8 as

$$RA_f(p'_{kl}, m') = \frac{1}{\sqrt{N}} \sum_{(i,j) \in L_{(p'_{kl}, m')}} f(i, j), \quad (4.9)$$

where  $N$  denotes the number of points lying on the line  $L_{(p'_{kl}, m')}$ .

Since for  $-\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4}$ , the slope  $m'$  varies between  $-1 \leq m' \leq 1$ . To make implementation easy, we restrict  $|m'| \leq 1$  and apply the Radon Transform as described in equation 4.9 on the image, and then we rotate the image 90 degrees and again apply the Radon Transform to the rotated image. In this way, we can compute  $RA_f(p'_{kl}, m')$  for every value of  $m'$ . Rotation at 90 degree is just like a changing the role of X and Y coordinates in the XY-plane, as discussed in [1].

The equation of line  $L_{(p'_{kl}, m')}$  can have the form

$$L_{(p'_{kl}, m')} = \begin{cases} \{(i, j) : j - l = m'(i - k)\}, & \text{unrotated Image} \\ \{(i, j) : m'(j - l) = i - k\}, & \text{rotated Image,} \end{cases} \quad (4.10)$$

or equivalently

$$L_{(p'_{kl}, \theta)} = \begin{cases} \{(x, y) : y - y'_l = (x - x'_k) \tan \theta\}, & \text{if } \theta \in [-\pi/4, \pi/4] \\ \{(x, y) : y - y'_l = (x - x'_k) \cot \theta\}, & \text{if } \theta \in (\pi/4, 3\pi/4) . \end{cases} \quad (4.11)$$

Since in a digital image the pixels values representing the straight line may not lie exactly on the straight line. Therefore, while computing the Radon coefficients in equation 4.9, our straight line  $L_{(p'_{kl}, m')}$ , as defined in equation 4.10 will be a thin strip instead of exact line.



Our implementation of the Radon Transform is closer to the Finite Radon Transform in the sense of summing the pixel values along a straight line. But our way of defining the straight line in equation 4.10 is different compared to the straight line defined in the Finite Radon Transform given in equation 4.3. The shift in defining a straight line helps us to overcome the problems associated with the Finite Radon Transform as discussed in [14] and [15].

## 4.2 Applications of Radon Transform

Radon Transform has wide range of applications especially in medical imaging and synthetic aperture radar imaging [12]. However, we have given its applications in linear edge detection and image approximation.

### 4.2.1 Detection of Linear Singularities in the image

For the linear edge detection, we present two closely related algorithms.

#### **Algorithm 1**

Let  $f(i, j)$  represents the pixel value of the point  $(i, j)$  in the image  $f$  of size  $m \times n$ . For simplicity we assume  $m = n$ . For this application, we don't need to apply the Radon Transform on the whole image. Instead, all possible Radon coefficients can be obtained just by applying Radon Transform on certain subset (intelligently chosen) of pixels in the image.

**STEP-1:** Define a set of points where the Radon Transform is to be applied to get Radon coefficients along all possible lines in the image. Since any line in the image will

hit exactly two boundaries of the image, we may pick a desired set  $S = \{(i, j) : (i, j) \in S1 \cup S2 \cup S3\}$ , where  $S1, S2$  and  $S3$  are the set of points along three boundaries of the image. Then any line in the image will pass through at least one of the points in the set  $S$ .

**STEP 2:** For the lines with  $|slope| \leq 1$ , apply the Radon Transform as discretized in equation 4.9 on the set of points  $S$  with the slope step size equal to  $\frac{1}{n}$ . This process will find Radon coefficients (normalized sum of pixel values along some specified line) along all the lines passing through points in  $S$  with slope between  $-1$  and  $1$ .

**STEP 3:** To find the Radon coefficients along the remaining lines in the image with  $|slope| > 1$ , rotate the image at angle of  $90$  (i.e. change the role of  $X$  and  $Y$  coordinates). In the rotated image, the slope of the remaining lines will become  $|slope| < 1$  and then apply step 2.

**STEP 4:** Store the Radon coefficients in separate classes so that the Radon coefficients corresponding to parallel lines are stored in the same class. Now for each class, if absolute value of the difference of two Radon coefficients corresponding to consecutive parallel lines is greater than the threshold  $T$  then linear singularity will be along the line  $L_b$  (either whole line or part of it) corresponding to the bigger Radon coefficient.

**STEP 5:** To find the exact position of linear singularity, compare each pixel value on the line  $L_b$  with its neighboring pixels in the image  $f$ . If the difference of the pixel value on  $L_b$  and average of its neighbors in  $f$  is significant (not zero for ideal case i.e. black and white image with linear features), then mark that point of the line as a singularity point. Then all the connected marked points will give a line segment corresponding to the linear singularity present in the image.

### Computational Cost

Computing Radon Transform in this algorithm is not costly. We need  $3n$  points in a

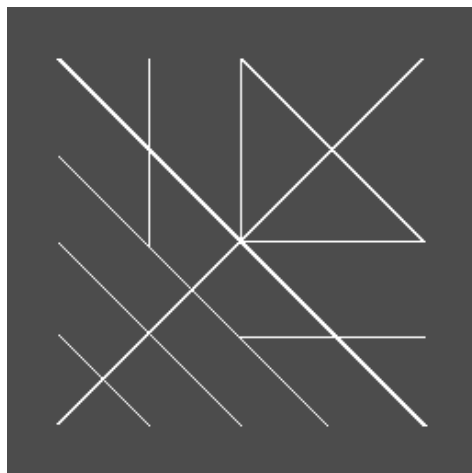


Figure 4.1: Simple image with linear singularities.

set  $S$  and for each point we have to compute multiplication  $3n$  times. Therefore, there are  $9n^2$  operations of multiplication. The complexity for computing the Radon Transform is  $O(n^2)$  for an image of size  $n^2$ .

### Algorithm 2

There may be many choices for the selection of set  $S$  in the **STEP 1** of above algorithm. We have only to make sure that all Radon coefficients corresponding to all possible lines in the image can be computed. The difference of this algorithm to the above algorithm is the choice of set  $S$  in **STEP 1**. One can chose a set  $S = \{(x, y) : (x, y) \in d_1 \cup d_2\}$ , where  $d_1$  is the set of points on the main diagonal of the image and  $d_2$  is the set of points on the anti diagonal to  $d_1$ . All other steps follows as are described in Algorithm 1. Computational cost of this algorithm is  $\frac{2}{3}$  of algorithm 1 because the number of points in the set  $S$  in algorithm 2 is  $\frac{2}{3}$  of the number of points in the set  $S$  of the algorithm 1.

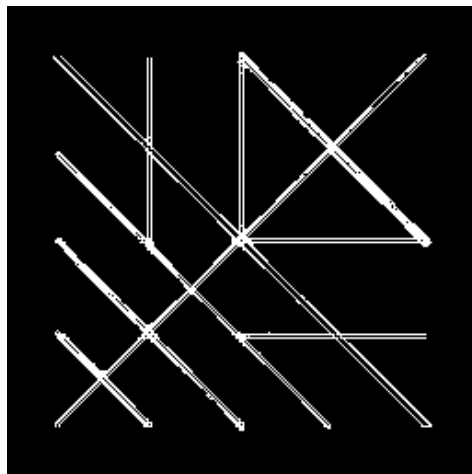


Figure 4.2: Detected Boundaries of linear singularities in the image given in figure 4.1.

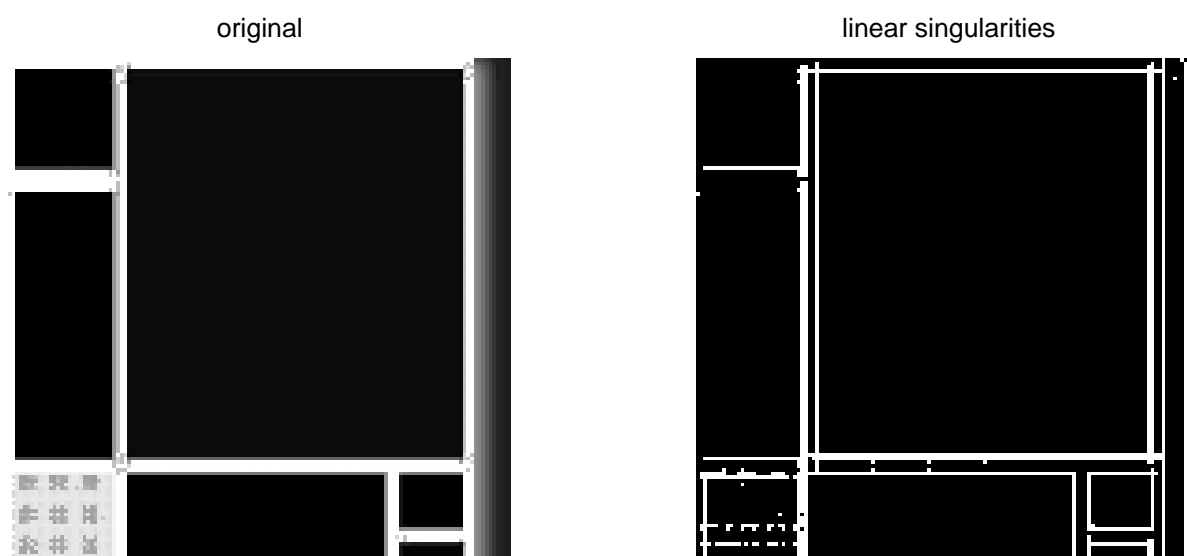


Figure 4.3: Image on the left and extracted linear singularities on the right.

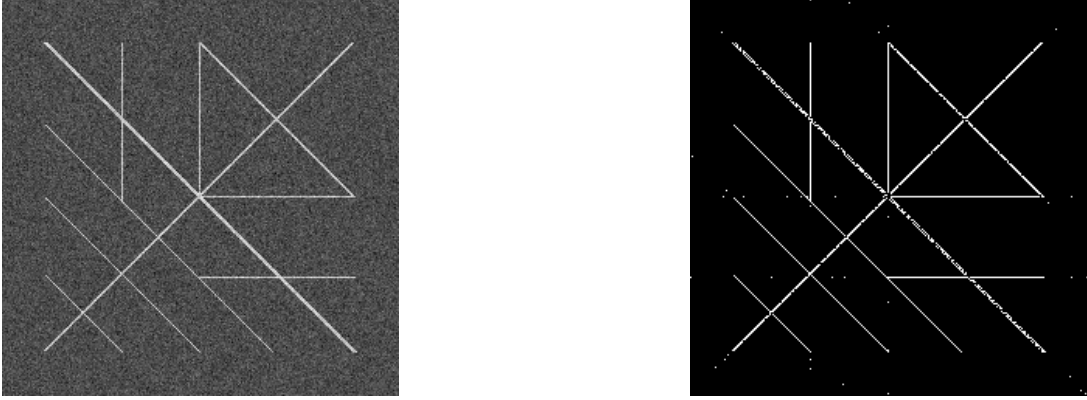


Figure 4.4: Noisy image on the left and extracted lines on the right.

#### 4.2.2 Image Approximation

Let  $f(i, j)$  represent a pixel value of the  $(i, j)$ th pixel in the image which also lies on the detected line  $L_b$  and  $av(i, j)$  is the average pixel value of those neighbors of  $(i, j)$  pixel which are not the part of line  $L_b$ . Now in the 4th step of the algorithm, instead of extracting the whole line  $L_b$  along the linear singularity, we extract the difference  $d(i, j) = f(i, j) - av(i, j)$  for each  $(i, j) \in L_b$  from the image. Similarly, we will extract all such differences along the detected lines where the linear singularities lie. The segmented image can now be written as

$$f = f_D + f_R$$

where

$$f_D(i, j) = \begin{cases} d(i, j), & \text{if } (i, j) \in L_b \\ 0, & \text{otherwise,} \end{cases} \quad (4.12)$$

and

$$f_R(i, j) = \begin{cases} av(i, j), & \text{if } (i, j) \in L_b \\ f(i, j), & \text{otherwise.} \end{cases} \quad (4.13)$$

This process will make the remaining image  $f_R$  (which is also called residual image)

smoother in the sense that all the linear singularities along lines or line segments are taken out and consequently sharpness of the linear edges decreases as compared to their sharpness in the original image.

We can apply the algorithm repeatedly on the residual image and can make the linear edges of the residual image less and less sharp. After the  $n^{\text{th}}$  application of the algorithm, we have

$$f = f_{D_1} + f_{D_2} + \cdots + f_{D_n} + f_R.$$

Now we have fragmented image, consisting of linear features in the form cartoon image  $f_{D_1} + f_{D_2} + \cdots + f_{D_n}$  and a residual image  $f_R$  which is relatively smoother than the original image  $f$ . Which is similar to the well known  $f = u+v$  model used in the image compression, where  $u$  is the cartoon image and  $v$  is the texture image. Usually the residual image is approximated by the wavelet method because it works well for the smoother images. The cartoon image can be approximated separately by using Wedgelet Transform [17] or any other transform which is optimal for the images with linear features. Approximation of the cartoon image and the residual image are added back to get the approximation of the original image.

### 4.3 Numerical Results

In our experiments for detecting linear features from the image, our Radon Transform scheme works very well. In figure 4.3, an image with linear features and the extracted linear singularities are shown. We applied this scheme on noisy images with linear features and it successfully extracts linear singularities. However, we have to modify the step 5 of algorithm 1 to deal with the noisy images. The significance in the difference of the pixel value on  $L_b$  and average of its neighbors in  $f$  in step 5 can vary with the noise level. Figure

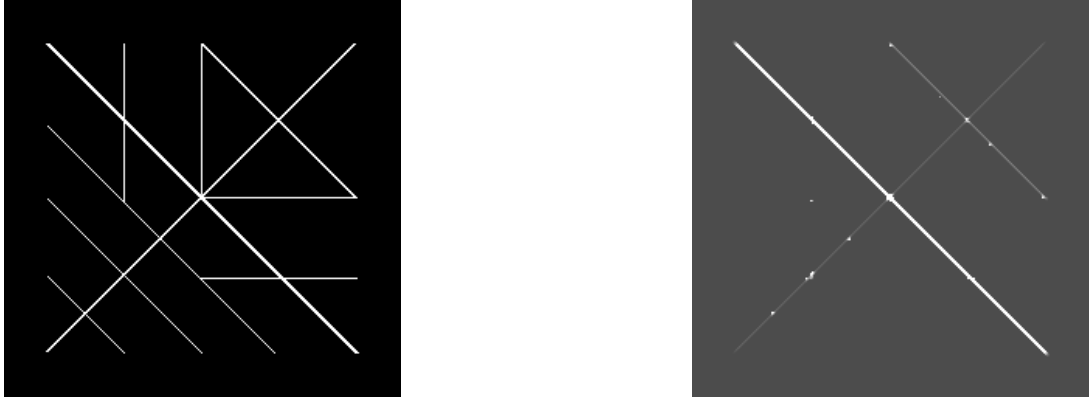


Figure 4.5: On the left is the cartoon image  $(f_{D_1} + f_{D_2} + f_{D_3} + f_{D_4})$  consists of the singularities extracted during 4 iterations and on the right is remaining smoother image  $f_R$ .



Figure 4.6: Original cameraman image.



Figure 4.7: On the left is the cartoon image consists of 346 linear singularities extracted during 4 iterations i.e.  $f_{D_1} + f_{D_2} + f_{D_3} + f_{D_4}$  and on the right is remaining smoother image  $f_R$ .

4.4, contains a noisy image and the extracted linear singularities. Figure 4.5 and figure 4.7 explain the separation of cartoon image and the residual image as an application of the Radon Transform.

With some modification in algorithm 1, we can determine the boundaries of linear features in the image. In step 4 of algorithm 1, if we choose the line  $L_b$  corresponding to the smaller Radon coefficient then the result will be a boundary of the linear singularities present in the image. Figure 4.2 contains the boundaries of the linear feature of the image given in figure 4.1.





# Appendix

In order to show that the transform  $\mathbf{A}$  smoothes the signal  $f$  in the sense of reduced 2nd difference norm. We consider a simple case of 3 by 3 transform matrix  $\mathbf{A}$  applied on the different kind of edges present in the image. Since

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ -a & 1 & a \\ 0 & 0 & 1 \end{pmatrix}$$

where  $a \in (0, 1)$ .

The 2-Norm of the matrix  $\mathbf{A}$  is defined by

$$\|\mathbf{A}\|_2^2 = \max|\lambda|,$$

where  $\lambda$ 's are the eigenvalues of  $\mathbf{A}^t \mathbf{A}$ .

$$\mathbf{A}^t \mathbf{A} = \begin{pmatrix} 1 + a^2 & -a & -a^2 \\ -a & 1 & a \\ -a^2 & -a & 1 + a^2 \end{pmatrix}$$

Thus  $|\mathbf{A}^t \mathbf{A} - \lambda I| = 0$  implies that  $\lambda = 1$  and  $\lambda = 1 + a^2 \pm \sqrt{a^2(a^2 + 2)}$ . Since  $0 < a < 1$ , therefore,  $\max|\lambda| = 1 + a^2 + \sqrt{a^2(a^2 + 2)}$ . Thus we have,

$$\|\mathbf{A}\|_2^2 = 1 + a^2 + \sqrt{a^2(a^2 + 2)}.$$

Since  $\|\mathbf{A}\|_\infty$  is defined as maximum of the row sum of the absolute values, therefore,

$$\|\mathbf{A}\|_\infty = 1 + a.$$

### 4.3.1 Modification of Piecewise Constant Function

Let  $f = (f_0, f_1, f_2, \dots, f_{N-1})$  be a given signal having singularity at the point  $M$ . Without loss of generality, we can assume that

$$f_j = \begin{cases} P & \text{if } 0 \leq j \leq M \\ R & \text{if } M < j \leq N - 1. \end{cases}$$

#### Case 1:- Modifying $f_M$

After applying smoothing transform  $A$  on the segment  $(f_{M-1}, f_M, f_{M+1})$  of the signal  $f$ , let the new signal is  $\tilde{f} = (f_0, f_1, f_2, \dots, f_{M-1}, f_M + a(f_{M+1} - f_{M-1}), f_{M+1}, \dots, f_{N-1}) = (P, P, \dots, P, P + a(R - P), R, \dots, R)$ . Then

$$\Delta^2 \tilde{f}_{M-1} = (R - P)a,$$

$$\Delta^2 \tilde{f}_M = (R - P)(1 - 2a),$$

and

$$\Delta^2 \tilde{f}_{M+1} = (R - P)(a - 1).$$

Therefore,

$$\begin{aligned} \sum_{j=M-1}^{M+1} |\Delta^2 \tilde{f}_j|^2 &= (a^2 + (1 - 2a)^2 + (a - 1)^2)(R - P)^2 \\ &= 2(1 - 3a(1 - a))(R - P)^2 \\ &< 2(R - P)^2, \end{aligned}$$

because  $1 - 3a(1 - a) < 1$  for  $0 < a < 1$ .

Now

$$\Delta^2 f_{M-1} = 0,$$

$$\Delta^2 f_M = R - P,$$

and

$$\Delta^2 f_{M+1} = P - R.$$

Thus

$$\sum_{j=M-1}^{M+1} |\Delta^2 f_j|^2 = 2(R - P)^2.$$

Since there is no singularity in  $f$  other than at the point  $M$  all the other values except  $f_M$  in  $\tilde{f}$  will remain unchanged, and therefore, we have,

$$\Delta^2 f_j = \Delta^2 \tilde{f}_j \text{ for } j \neq M - 1, M, M + 1.$$

Thus we have,

$$\sum_j |\Delta^2 \tilde{f}_j|^2 < \sum_j |\Delta^2 f_j|^2.$$

### Case 2:- Modifying $f_{M+1}$

To modify  $f_{M+1}$ , we need to apply inverse of transform  $A$  on the segment  $(f_M, f_{M+1}, f_{M+2})$  of the signal  $f$ , let the new signal is  $\tilde{f} = (f_0, f_1, f_2, \dots, f_M, f_{M+1} - a(f_{M+2} - f_M), f_{M+2}, \dots, f_{N-1}) = (P, P, \dots, P, R - a(R - P), R, \dots, R)$ . Thus

$$\Delta^2 \tilde{f}_M = (R - P)(1 - a),$$

$$\Delta^2 \tilde{f}_{M+1} = (R - P)(-1 + 2a),$$

and

$$\Delta^2 \tilde{f}_{M+2} = (R - P)(-a).$$

Therefore,

$$\begin{aligned} \sum_{j=M}^{M+2} |\Delta^2 \tilde{f}_j|^2 &= ((-a)^2 + (-1 + 2a)^2 + (1 - a)^2)(R - P)^2 \\ &= 2(1 - 3a(1 - a))(R - P)^2 \\ &< 2(R - P)^2. \end{aligned}$$

While

$$\Delta^2 f_M = R - P,$$

$$\Delta^2 f_{M+1} = P - R,$$

and

$$\Delta^2 f_{M+2} = 0.$$

Therefore,

$$\sum_{j=M}^{M+2} |\Delta^2 f_j|^2 = 2(R - P)^2.$$

Since there is no singularity in  $f$  other than at the point  $M$ , so all the other values except  $f_{M+1}$  in  $\tilde{f}$  will remain unchanged and therefore, we have,

$$\Delta^2 f_j = \Delta^2 \tilde{f}_j \text{ for } j \neq M, M + 1, M + 2.$$

Thus we have,

$$\sum_j |\Delta^2 \tilde{f}_j|^2 < \sum_j |\Delta^2 f_j|^2.$$

### 4.3.2 Modification of Piecewise Smooth Function

Let  $f = (f_0, f_1, f_2, \dots, f_{N-1})$  be a given piecewise smooth signal having singularity at the point  $M$  (i.e.  $|f_M - f_{M+1}| \geq T$ ). Where  $T$  is some threshold.

Without loss of generality, we can assume that

$$f_M = f_{M-1} + s_1(\Delta x),$$

$$f_{M-2} = f_{M-1} + s_2(\Delta x),$$

$$f_{M+2} = f_{M+1} + s_3(\Delta x),$$

and

$$f_{M+3} = f_{M+1} + s_4(\Delta x),$$

where  $s_1(\Delta x)$ ,  $s_2(\Delta x)$ ,  $s_3(\Delta x)$  and  $s_4(\Delta x)$  are very small quantities and depend upon the local smoothness of  $f$ .

### Case 1:- Modifying $f_M$

After applying smoothing transform  $A$  on the segment  $(f_{M-1}, f_M, f_{M+1})$  of the signal  $f$ , let the new signal is  $\tilde{f} = (f_0, f_1, f_2, \dots, f_{M-1}, f_M + a(f_{M+1} - f_{M-1}), f_{M+1}, \dots, f_{N-1})$ . Thus,

$$\Delta^2 \tilde{f}_{M-1} = \Delta^2 f_{M-1} + a(f_{M+1} - f_{M-1}),$$

$$\Delta^2 \tilde{f}_M = \Delta^2 f_M - 2a(f_{M+1} - f_{M-1}),$$

and

$$\Delta^2 \tilde{f}_{M+1} = \Delta^2 f_{M+1} + a(f_{M+1} - f_{M-1}).$$

Since there is no singularity in  $f$  other than at the point  $M$ , so all the other values except  $f_M$  in  $\tilde{f}$  will remain unchanged and therefore, we have,

$$\Delta^2 f_j = \Delta^2 \tilde{f}_j \text{ for } j \neq M - 1, M, M + 1.$$

Let  $J = (f_{M+1} - f_{M-1})$ , then we have,

$$\sum_j |\Delta^2 \tilde{f}_j|^2 = \sum_j |\Delta^2 f_j|^2 + 6a^2 J^2 + 2aJ(\Delta^2 f_{M-1} + \Delta^2 f_{M+1} - 2\Delta^2 f_{M-1}).$$

But

$$\begin{aligned} \Delta^2 f_{M-1} + \Delta^2 f_{M+1} - 2\Delta^2 f_{M-1} &= f_{M-2} - 4f_{M-1} + 6f_M - 4f_{M+1} + f_{M+2} \\ &= f_{M-1} + s_2(\Delta x) - 4f_{M-1} + 6f_{M-1} \\ &\quad + 6s_1(\Delta x) - 4f_{M+1} + f_{M+1} + s_3(\Delta x) \\ &= -3J + 6s_1(\Delta x) + s_2(\Delta x) + s_3(\Delta x). \end{aligned}$$

Therefore,

$$\begin{aligned} \sum_j |\Delta^2 \tilde{f}_j|^2 &= \sum_j |\Delta^2 f_j|^2 + 6a^2 J^2 + 2aJ[-3J + 6s_1(\Delta x) + s_2(\Delta x) + s_3(\Delta x)] \\ &= \sum_j |\Delta^2 f_j|^2 - 6a(1-a)J^2 + 2aJ[6s_1(\Delta x) + s_2(\Delta x) + s_3(\Delta x)] \\ &= \sum_j |\Delta^2 f_j|^2 - [6a(1-a)J^2 - 2aJ(6s_1(\Delta x) + s_2(\Delta x) + s_3(\Delta x))]. \end{aligned}$$

Since  $J$  is equivalent to the size of the jump present in the signal and we are only modifying the jumps of large magnitude, therefore, its magnitude is very very big as compared to  $s_1(\Delta x)$ ,  $s_2(\Delta x)$  and  $s_3(\Delta x)$ . Thus

$$[6a(1-a)J^2 - 2aJ(6s_1(\Delta x) + s_2(\Delta x) + s_3(\Delta x))] > 0,$$

and consequently

$$\sum_j |\Delta^2 \tilde{f}_j|^2 < \sum_j |\Delta^2 f_j|^2.$$

### Case 2:- Modifying $f_{M+1}$

To modify  $f_{M+1}$ , we need to apply inverse of transform  $A$  on the segment  $(f_M, f_{M+1}, f_{M+2})$  of the signal  $f$ , let the new signal is  $\tilde{f} = (f_0, f_1, f_2, \dots, f_M, f_{M+1} - a(f_{M+2} - f_M), f_{M+2}, \dots, f_{N-1})$ . Thus,

$$\begin{aligned}\Delta^2 \tilde{f}_M &= \Delta^2 f_M - a(f_{M+1} - f_{M-1}), \\ \Delta^2 \tilde{f}_{M+1} &= \Delta^2 f_{M+1} + 2a(f_{M+1} - f_{M-1}),\end{aligned}$$

and

$$\Delta^2 \tilde{f}_{M+2} = \Delta^2 f_{M+2} - a(f_{M+1} - f_{M-1}).$$

Since there is no singularity in  $f$  other than at the point  $M$ , so all the other values except  $f_{M+1}$  in  $\tilde{f}$  will remain unchanged and therefore, we have,

$$\Delta^2 f_j = \Delta^2 \tilde{f}_j \text{ for } j \neq M, M+1, M+2.$$

Let  $J = (f_{M+1} - f_{M-1})$ , then we have,

$$\sum_j |\Delta^2 \tilde{f}_j|^2 = \sum_j |\Delta^2 f_j|^2 + 6a^2 J^2 + 2aJ(-\Delta^2 f_M + 2\Delta^2 f_{M+1} - \Delta^2 f_{M+2}).$$

But

$$\begin{aligned}-\Delta^2 f_M + 2\Delta^2 f_{M+1} - \Delta^2 f_{M+2} &= -f_{M-1} + 4f_M - 6f_{M+1} + 4f_{M+2} - f_{M+3} \\ &= -f_{M-1} + 4f_{M-1} + 4s_1(\Delta x) - 6f_{M+1} + 4f_{M+1} \\ &\quad + 4s_3(\Delta x) - f_{M+3} - s_4(\Delta x) \\ &= -3J + 4s_1(\Delta x) + 4s_3(\Delta x) - s_4(\Delta x).\end{aligned}$$

Therefore,

$$\begin{aligned}\sum_j |\Delta^2 \tilde{f}_j|^2 &= \sum_j |\Delta^2 f_j|^2 + 6a^2 J^2 + 2aJ[-3J + 4s_1(\Delta x) + 4s_3(\Delta x) - s_4(\Delta x)] \\ &= \sum_j |\Delta^2 f_j|^2 - 6a(1-a)J^2 + 2aJ[4s_1(\Delta x) + 4s_3(\Delta x) - s_4(\Delta x)]\end{aligned}$$



$$= \sum_j |\Delta^2 f_j|^2 - [6a(1-a)J^2 - 2aJ(4s_1(\Delta x) + 4s_3(\Delta x) - s_4(\Delta x))].$$

Since  $J$  is equivalent to the size of the jump present in the signal and we are only modifying the jumps of large magnitude, therefore, its magnitude is very very big as compared to  $s_1(\Delta x)$ ,  $s_3(\Delta x)$  and  $s_4(\Delta x)$ . Thus

$$[6a(1-a)J^2 - 2aJ(4s_1(\Delta x) + 4s_3(\Delta x) - s_4(\Delta x))] > 0,$$

and consequently

$$\sum_j |\Delta^2 \tilde{f}_j|^2 < \sum_j |\Delta^2 f_j|^2.$$

# Bibliography

- [1] A.Averbuch, R.R.Coifman, D.L.Donoho, M.Israeli, J.Walden, Fast Slant Stack: A notion of Radon Transform for Data in a Cartesian Grid which is Rapidly computible, Algebraically Exact, Geometrically Faithful and Invertible, A Technical Report, Department of Statistics, Stanford University, 2001.
- [2] Charles. K. Chui, An introduction to wavelets, Academic Press, Boston, 1992.
- [3] C. Gasquet and P. Witomski, Fourier Analysis and Applications, Springer-Verlag New York, Inc. NY 1998.
- [4] David L. Donoho, Orthonormal Ridgelets and Linear Singularities, J. Mathematical Analysis, vol.31, No.5(1062-1099), 2000.
- [5] David L. Donoho, Wedgelets: Nearly-Minimax Estimation of Edges, Technical Report, Department of Statistics, Stanford University, Stanford, CA, 1997.
- [6] Emmanuel J. Candes and David L. Donoho, Ridgelets: A key to higher-dimensional intermittency?, R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci., vol. 357 (2495-2509), 1999.
- [7] Emmanuel J. Candes and David L. Donoho, Curvelets - a surprisingly effective non-adaptive representation for objects with edges, in Curves and Surfaces, L.L. et al. Schumaker, Ed., (Vanderbilt University Press), Nashville, TN, 1999.

- [8] Frantisek Matus and Jan Flusser, Image Representations via a Finite Radon Transform, *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 15, No. 10, October 1993.
- [9] Gilbert Strang and Truong Nguyen, *Wavelets and Filter banks*, Wellesley - Cambridge Press, Wellesley, MA, 1996.
- [10] G.T.Herman, *Image Reconstruction from Projections, The Fundamentals of Computerized Tomography*, Academic Press, 1980.
- [11] Ingrid Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, PA, 1992.
- [12] Jean-Luc Starck, Emmanuel J. Candes and David L. Donoho, The Curvelet Transform for Image Denoising, *IEEE transactions on image processing*, vol.11, NO.6, June 2002.
- [13] Martin Vetterli and Jelena Kovačević, *Wavelets and Subband Coding*, Prentice-Hall Inc., NJ, 1995.
- [14] Minh N. Do and Martin Vetterli, Orthonormal Finite Ridgelet Transform for Image Compression, *ICIP (International Conference for Image Processing) Canada*, September 2000.
- [15] Minh N. Do and Martin Vetterli, The finite Ridgelet transform for image compression, *IEEE Transactions on Image Processing*, January 2003.
- [16] Michael Wakin and Justin Romberg, Geometric Tools for image compression, *Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, (Accepted, November, 2002).
- [17] M. B. Wakin, J. K. Romberg, H. Choi, and R. G. Baraniuk, "Image compression using an efficient edge cartoon + texture model," in *Proc., IEEE Data Compression Conference*, Snowbird, Utah, April 2002, (43-52).

- [18] Martin L. Brady, A Fast Discrete Approximation Algorithm For The Radon Transform, SIAM J. COMPUT. vol.27, NO.1, (107-119), February 1998.
- [19] P. Claypoole, G. Davis, W. Sweldens and R. Baraniuk, Nonlinear Wavelet Transforms for Image Coding via Lifting, IEEE Trans. Image Process., vol.12, No. 12 (1449-1459), December 2003.
- [20] Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, Second Edition, Pearson Education Inc., 2002.
- [21] S. Mallat, A Wavelet Tour of Signal Processing, Academic Press, San Diego, CA, 1998.
- [22] Tony F. Chan and Jianhong (Jackie) Shen, Image Processing and Analysis - Variational, PDE, wavelet, and stochastic methods, SIAM, Philadelphia, PA, 2005.
- [23] Tony F. Chan and H. M. Zhou, ENO-Wavelet Transforms for Piecewise Smooth Functions, SIAM J. Numer. Anal., vol.40, No. 4 (1369-1404), 2002.