Query of image content using Wavelets and Gibbs-Markov Random Fields

Imtiaz Hossain
Louisiana State University and Agricultural and Mechanical College, imtiaz@math.lsu.edu

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_theses

Part of the Applied Mathematics Commons

Recommended Citation
https://digitalcommons.lsu.edu/gradschool_theses/655

This Thesis is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Master's Theses by an authorized graduate school editor of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.
QUERY BY IMAGE CONTENT
USING WAVELETS AND GIBBS-MARKOV RANDOM FIELDS

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science
in

The Department of Mathematics

by
Imtiaz Hossain
B.C.A., Devi Ahilya University, Indore, India, 1998
August 2004
To Dadabhai, Dida and Dadi. This one is for you.
Acknowledgements

I have a lot to be thankful for. My sincerest thanks and gratitude go out to:

Dr. Gestur Olafsson, my thesis advisor, for his continued guidance and help. He has been the main motivating factor behind my decision to pursue the Masters program in Applied Mathematics at LSU.

Dr. Vibhas Aravamuthan, for his guidance, support, advice and the learning opportunities he provided throughout my stay at LSU.

Dr. George Cochran, for all his help with the proofs in the section on Markov Random Fields, and for being on my thesis committee. Thanks also for being such a terrific instructor.

I would like to thank Dr. Roy K. Dokka for his continued encouragement and support. I am thankful to all my professors in the Department of Mathematics and the Department of Electrical Engineering for a wonderful learning experience. Thanks to all my friends at LSU for making my stay memorable.

Finally, a big thank you to my family for all their love and support. I can’t thank you enough.
# Table of Contents

Acknowledgements ........................................................................ iii

List of Figures ........................................................................... vi

Abstract ....................................................................................... x

1 Introduction ............................................................................. 1
  1.1 Annotation-based Image Query ................................................ 1
  1.2 Query by Image Content .......................................................... 2

2 Transforms for Image Analysis .................................................. 5
  2.1 Fourier Transform ................................................................. 5
  2.2 Variations of the Fourier Transform ................................ .......... 6
  2.3 Gabor Transforms .................................................................. 8
  2.4 The Wavelet Transform ......................................................... 8
    2.4.1 The Continuous Wavelet Transform ..................................... 9
    2.4.2 The Discrete Wavelet Transform ........................................ 12
    2.4.3 Bases and Frames ............................................................. 13
    2.4.4 Multiresolution ................................................................. 16
    2.4.5 The Haar Wavelet .............................................................. 23

3 Multiresolution Content Based Image Query ................................. 30
  3.1 The Wavelet Metric ............................................................... 31
  3.2 Training the Weights ............................................................. 32
  3.3 The Logit Model for Least Squares .......................................... 33
  3.4 Quantization ........................................................................... 33
  3.5 Textures .................................................................................. 34
    3.5.1 Forcing Detail Coefficients ................................................... 37
    3.5.2 Windowing ......................................................................... 37
  3.6 Results and Implementation Details ......................................... 38

4 Markov Random Fields in Image Processing ................................. 48
  4.1 Introduction ............................................................................ 48
  4.2 Random Fields ....................................................................... 48
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.1</td>
<td>Models and Energy Functions</td>
<td>50</td>
</tr>
<tr>
<td>4.2.2</td>
<td>The Linear Model</td>
<td>50</td>
</tr>
<tr>
<td>4.2.3</td>
<td>The Degradation Model</td>
<td>51</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Stochastic Models</td>
<td>51</td>
</tr>
<tr>
<td>4.2.5</td>
<td>Pseudo-Joint Probability</td>
<td>52</td>
</tr>
<tr>
<td>4.2.6</td>
<td>MAP</td>
<td>53</td>
</tr>
<tr>
<td>4.3</td>
<td>Neighborhoods</td>
<td>54</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Neighborhood Order</td>
<td>56</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Cliques</td>
<td>56</td>
</tr>
<tr>
<td>4.4</td>
<td>Gibbs Random Fields</td>
<td>58</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Clique Potentials</td>
<td>60</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Local Conditional Probabilities for a GRF</td>
<td>61</td>
</tr>
<tr>
<td>4.5</td>
<td>Markov Random Fields</td>
<td>62</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Hammersley-Clifford Theorem</td>
<td>63</td>
</tr>
<tr>
<td>4.5.2</td>
<td>The Exponential Family</td>
<td>66</td>
</tr>
<tr>
<td>4.5.3</td>
<td>From Gibbs to Other Distributions</td>
<td>67</td>
</tr>
<tr>
<td>4.5.4</td>
<td>Models and Energy Functions</td>
<td>69</td>
</tr>
<tr>
<td>4.6</td>
<td>Parameter Estimation</td>
<td>70</td>
</tr>
<tr>
<td>4.6.1</td>
<td>Least Squares</td>
<td>71</td>
</tr>
<tr>
<td>4.6.2</td>
<td>Conditional Least Squares</td>
<td>72</td>
</tr>
<tr>
<td>4.6.3</td>
<td>Simulated Annealing</td>
<td>73</td>
</tr>
<tr>
<td>4.7</td>
<td>Applications to Image Processing</td>
<td>73</td>
</tr>
<tr>
<td>4.7.1</td>
<td>Smoothing</td>
<td>73</td>
</tr>
<tr>
<td>4.7.2</td>
<td>Texture Generation</td>
<td>81</td>
</tr>
<tr>
<td>4.8</td>
<td>Pattern Matching and Search Using Wavelet Multiresolution</td>
<td>83</td>
</tr>
<tr>
<td>4.9</td>
<td>Conclusion and Future Direction</td>
<td>87</td>
</tr>
</tbody>
</table>

Bibliography | 88 |
Vita | 91 |
List of Figures

1.1 Some results from a search of ”yellow flower” on www.google.com .................. 2
1.2 How would one search for such patterns using an annotation-based search? .......... 2
1.3 Global nature of commonly used metrics. .................................................. 3
2.1 Dilates of the Haar wavelet function ...................................................... 18
2.2 Multiresolution Spaces ........................................................................ 20
2.3 The Haar Wavelet Function .................................................................... 23
2.4 The Haar Scaling Function ...................................................................... 24
2.5 A sample 1D signal .................................................................................. 24
2.6 Haar Wavelet Transform applied to the 1D signal .................................... 25
2.7 Wavelet Decomposition .......................................................................... 26
2.8 Wavelet Reconstruction ........................................................................... 27
2.9 A sample image ......................................................................................... 28
2.10 The Haar Wavelet Transform applied on the rows of the previous image.......... 28
2.11 The Haar Wavelet Transform on the columns of the row-transformed image..... 29
3.1 Obtaining the wavelet signature ............................................................... 30
3.2 Example layout of the selection of signature coefficients ............................ 33
3.3 \( L(x) = \log \frac{x}{1-x} \) for \( x \in [0, 1] \) ........................................ 34
3.4 \( \frac{x}{1-x} \) for \( x \in [0, 1] \) ........................................ 35
3.5 Some sample textures from the VISTEX database. ....................... 36
3.6 Limiting the algorithm to specific levels .................................. 37
3.7 Positional Matches ...................................................... 37
3.8 Using the Windowing Technique ........................................ 38
3.9 Query Image 1 ........................................................... 40
3.10 Ranking of Images from the Target Database, w.r.t. Query 1 .......... 40
3.11 Target Image with the lowest metric-rank, w.r.t. Query 1 ............... 40
3.12 Query Image 2 ........................................................... 41
3.13 Ranking of Images from the Target Database, w.r.t. Query 2 .......... 41
3.14 Target Image with the lowest metric-rank, w.r.t. Query 1 ............... 41
3.15 Query Image 3 ........................................................... 42
3.16 Ranking of Images from the Target Database, w.r.t. Query 3 .......... 42
3.17 Target Image with the lowest metric-rank, w.r.t. Query 3 ............... 42
3.18 Query Image 4 ........................................................... 43
3.19 Ranking of Images from the Target Database, w.r.t. Query 4 .......... 43
3.20 Target Image with the lowest metric-rank, w.r.t. Query 4 ............... 43
3.21 Query Image 5 ........................................................... 44
3.22 Ranking of Images from the Target Database, w.r.t. Query 5 .......... 44
3.23 Target Image with the lowest metric-rank, w.r.t. Query 5 ............... 44
3.24 Query Image 6 ........................................................... 45
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.15</td>
<td>Gaussian filter in figure 4.14 applied to noisy image.</td>
</tr>
<tr>
<td>4.16</td>
<td>$5 \times 5$ Gaussian smoothing filter with $\mu = 0$ and $\sigma = 1$.</td>
</tr>
<tr>
<td>4.17</td>
<td>Gaussian filter in figure 4.16 applied to noisy image.</td>
</tr>
<tr>
<td>4.18</td>
<td>$5 \times 5$ Gaussian smoothing filter with $\mu = 0$ and $\sigma = 2$.</td>
</tr>
<tr>
<td>4.19</td>
<td>Gaussian filter in figure 4.18 applied to noisy image.</td>
</tr>
<tr>
<td>4.20</td>
<td>MRF-based Filtering, Pass 1. (Error Norm $S_1 = 406.5412$)</td>
</tr>
<tr>
<td>4.21</td>
<td>MRF-based Filtering, Pass 2. (Error Norm $S_2 = 21.96$)</td>
</tr>
<tr>
<td>4.22</td>
<td>MRF-based Filtering, Pass 3. (Error Norm $S_3 = 10.798$)</td>
</tr>
<tr>
<td>4.23</td>
<td>MRF-based Filtering, Pass 4. (Error Norm $S_4 = 4.7281$)</td>
</tr>
<tr>
<td>4.24</td>
<td>Texture Generation from MRF parameter values</td>
</tr>
<tr>
<td>4.25</td>
<td>Texture Generation from MRF parameter values</td>
</tr>
<tr>
<td>4.26</td>
<td>Original VISTEX texture images</td>
</tr>
<tr>
<td>4.27</td>
<td>Matching Images 1 through 4</td>
</tr>
<tr>
<td>4.28</td>
<td>Matching Images 13 through 16</td>
</tr>
<tr>
<td>4.29</td>
<td>Matching Images 25 through 28</td>
</tr>
<tr>
<td>4.30</td>
<td>Matching Images 69 through 72</td>
</tr>
</tbody>
</table>
Abstract

The central theme of this thesis is the application of Wavelets and Random Processes to content-based image query (on texture patterns, in particular). Given a query image, a content-based search extracts a certain representative measure (or signature) from the query image and likewise for all the target images in the search archive. A good representative measure is one that provides us with the ability to differentiate easily between different patterns. A distance measure is computed between the query properties and the properties of each of the target images. The lowest distance measure gives us the best target match for the particular query. Typically, the measure extraction on the target archive is performed as a pre-processing step.

The thesis features two different methods of measure extraction. The first one is a wavelet based measure extraction method. It builds upon a previously documented method, but adds subtle modifications to it so that it now lends much more effectiveness to pattern matching on texture patterns and on images of unequal sizes. The modified algorithm as well as the mathematics behind it is presented. The second method uses a Markov Random Field to model the texture properties of regions within an image. The parameters of the model serve as the texture measure or signature. Wavelet-based multiresolution is then used to speed up the search. The theory of Markov Random Fields, their equivalence with Gibbs Random Fields, the Hammerseley-Clifford theorem and parameter estimation techniques are presented. In addition to pattern matching these texture signatures have also be used for controlled image smoothing and texture generation. The results from both methods are encouraging. One hopes that these methods find widespread use in image query applications.
Chapter 1

Introduction

The main theme of this thesis is image retrieval. The central idea is to be able to search through an archive of images and retrieve only those that satisfy a pre-specified criteria. There are two main categories of image searches:

- Annotation-based image query
- Content-based image query

For example, one can log on to a web-based search engine, say www.google.com and type in keywords, say *flower*, *bicycle*, etc. and the search engine pulls up seemingly relevant target images. This is an example of Annotation-based search.

1.1 Annotation-based Image Query

In this kind of image query an image match is achieved by matching textual keywords that have been previously assigned to each image. For example in web-based searches, the name of the image file forms an annotation and so do the contents of the page within which the image is embedded. A good example of an Annotation based approach is a medical DICOM database. The Physician annotates the pathology and they are stored as part of the patient’s record. A search on pathology characteristics would then pull up the respective record.

But not every image is annotated so carefully. Going back to the web based search engine, each image is deemed to be annotated with keywords that may be words that appear on the same page as the image or the name of the file. So if an image of a building were called *flower.jpg*, the search for *flower* would pull up the image of the building. Equally bad is the case where an image of a flower is called something else, like *nice.jpg*. A search for *flower* would not pull up this particular image.

An annotation based search is restrictive. The success of the search depends upon the choice of keywords used to describe each image. Very often one looks for a higher level of description when looking for a particular target image. For example one may be looking for an image with a blue background which has a yellow flower somewhere in the top left hand corner, a purple disc in the...
center and a pencil at the bottom. Such a high level of annotation detail is *not* ordinarily available for *every* image available on the web or other image archives in general.

![Figure 1.2: Some results from a search of "yellow flower" on www.google.com](image)

**Figure 1.2:** How would one search for such patterns using an annotation-based search?

### 1.2 Query by Image Content

A good alternative is to search for images by specifying its content graphically. For example, given a piece of a texture pattern, one may wish to find all images that contain such a texture. Or, given an outline diagram, one may wish to retrieve all images that contain objects of the form outlined in the diagram. Thus the central idea here is pattern matching. Given two image regions (possibly of unequal dimensions), the task of pattern matching is to find a robust measure of the difference between the two regions, if they are equal or whether one can be considered as be part of the other.
Hereafter any image region that forms the search criteria will be known as the *query* image. Every image (or a part thereof) from the archive that is being compared to the query image will be called the *target* image.

Let $Q$ denote the query image and $T$ denote the target image. An intuitive approach is to apply a pixel-by-pixel comparison. If the *query* and *target* images are of the same size, one may use a metric of comparison such as:

- **Mean of the absolute differences.**
  
  $$M(Q, T) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} |Q(i, j) - T(i, j)| \quad ; m, n, i, j \in \mathbb{Z}$$

- **Mean Square Error.**
  
  $$V(Q, T) = \sqrt{\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} [Q(i, j) - T(i, j)]^2} \quad ; m, n, i, j \in \mathbb{Z}$$

An absolute match yields a value of zero in both of the above cases. Relatively smaller values of both metrics may signify a similarity between the query and the target. But there is no guarantee since both are metrics are global measures. In other words, these measures do not incorporate any position-dependent information and hence do not model local behaviour well.

For example, in Figure 1.3, clearly $M(Q, T_1) = M(Q, T_2) = M(Q, T_3) = M(Q, T_4) = 1.5$ and $V(Q, T_1) = V(Q, T_2) = V(Q, T_3) = V(Q, T_4) = \sqrt{5}$, yet clearly $T_1 \neq T_2 \neq T_3 \neq T_4$.

![Figure 1.3: Global nature of commonly used metrics.](image)

The query and target image regions may be of different sizes (even different shapes). In such cases, if possible, the larger region is reduced to the size of the smaller region or the smaller region...
is matched with all possible regions of the same size within the larger region. Uniform color variations in the image acquisition process can be discounted by computing the zero-mean images of the respective query and target images.

For query and target images of the same size, the pixel-by-pixel approach involves at least $O(mn)$ computations where the $m \times n$ is the image size. For example, each of the above metrics involve $O(16384)$ operations for a set of two $128 \times 128$ images, $O(262144)$ operations for a set of two $512 \times 512$ images, $O(1048576)$ operations for a set of two $1024 \times 1024$ images. A target archive may contain thousands of images. Clearly such computations quickly become very prohibitive with an increase in the size of the image or the number of images in the archive.

The alternative is to devise an alternate representation for each image that

- has very few components as compared to those of original image $(mn)$.
- is fairly representative of the image.

Ideally we want the components to be able to uniquely identify an image. But since this representation uses a far less number of components, a certain amount of ambiguity is introduced. In this thesis we are mainly concerned with applications where a certain amount of ambiguity is allowable, i.e. we are not looking to distinguish between target images that vary at a few odd pixels (e.g. the variation may be due to noise). This representation is also known as a measure (or signature). A bijective map between the image space and the measure or signature space would be ideal, but is clearly not possible with further restrictions on either or both of these spaces. Instead what we would like to obtain are distinct equivalence classes of image patterns, each of which is described by a unique signature. Intuitively, the smaller the equivalence class, the more effective is the signature. But at the same time the components of the signature increase in size with a corresponding increase in computation while performing matches. This leaves one with a largely empirical decision as to the balance between the performance (ability to distinguish between different image patterns) of the algorithm and its speed. Restrictions on the image space may be applied to boost the performance of the algorithm. Chapter 3 of this thesis discusses a wavelet based measure, that was proposed in [CEJS95], which is used to match different images. Modifications to the algorithm are discussed, that yield much better results when comparing textures as opposed to images containing homogeneous regions.

The next chapter reviews the mathematics of the image analysis tools that we will require in the subsequent sections of this thesis. In particular, it covers the wavelet transforms which form the basis for the image query algorithm described in chapter 3.
Chapter 2

Transforms for Image Analysis

Throughout this thesis an image will be treated as a two dimensional discrete valued signal. Each row or column (or any other set of pixels for that matter, although rows and columns are more intuitive and useful) can be viewed as a discrete valued signal. Thus the analysis of an image can performed at a lower level, e.g. the individual rows and columns. An image is a varying pattern of color values. The location of an object in an image is characterized by the occurrence of a characteristic set of pixel values some location. A non-homogenous object has more intensity fluctuations that a homogenous object. Thus the main aim of the analysis is to find different types of variation (frequencies) and their location within the image.

2.1 Fourier Transform

A popular and powerful tool for detecting frequency information is the Fourier Transform. We try to represent a function $f(x)$ in terms of a set of basis functions of the type $e^{2\pi isx}$, i.e. simple waves.

**Definition 2.1.1** The Fourier Transform

The Fourier Transform of a function $f \in L^1(\mathbb{R})$ is defined (see [Kam00, HW89]) as

$$\hat{f}(s) = \int f(x) e^{-2\pi isx} dx$$

Note, however, that $\hat{f}(s)$ is not always in $L^1(\mathbb{R})$. For example:

$$f(x) = \begin{cases} 1, & -\frac{1}{2} < x < \frac{1}{2} \\ 0, & \text{otherwise.} \end{cases}$$

$$\hat{f}(s) = \text{sinc}(s) = \frac{\sin(\pi s)}{\pi s} \notin L^1(\mathbb{R})$$

Now, let $f \in L^2(\mathbb{R})$. It is known that $L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ is dense in $L^2(\mathbb{R})$. We can find a sequence $\{f_n\}$ of functions in $L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ such that $\|f_n - f\|_2 \xrightarrow{\ n \to \infty\ } 0$. One can show that $\hat{f}_n$ converges in the $L^2(\mathbb{R})$-norm to some function $\hat{f}$ which is called the Fourier Transform of $f$. Here

$$\hat{f}_n(s) = \int f_n(x) e^{-2\pi isx} dx$$
Now, $L^2(\mathbb{R}, \mu)$ is a Hilbert space with the usual canonical scalar product $\langle \cdot, \cdot \rangle$ and where $\mu$ is the Lebesgue measure on $\mathbb{R}$. Let $e_s(x) = e^{2\pi i sx}$. We know that $e_s \notin L^2(\mathbb{R})$ and thus $\hat{f}$ cannot always be represented as in the scalar product form $\langle f, e_s \rangle$. In other words, one cannot say for sure if the pairing is well defined. Now let $e_s = e_s \chi_{[-n, n]}$, $n \in \mathbb{R}$ where $\chi_{[-n, n]}$ is the indicator function on the interval $[-n, n]$. So $e_s$ has compact support on $\mathbb{R}$ and is clearly an $L^2$ function. We also know that if $f \in L^2(\mathbb{R})$ then $f \chi_{[-n, n]}$ is also in $L^2(\mathbb{R})$. So $\langle f, e_s \rangle$ is well defined and

$$
\langle f, e_s \rangle = \int_{\mathbb{R}} f(x) e_s(x) \chi_{[-n, n]}(x) \, dx
= \int_{\mathbb{R}} f(x) \chi_{[-n, n]}(x) e_s(x) \, dx
= \langle f \chi_{[-n, n]}, e_s \rangle
$$

Thus the sequence $\{f_n\}$ may be chosen as follows (see [HW89]):

$$f_n = f \chi_{[-n, n]}$$

### 2.2 Variations of the Fourier Transform

**Case 1. $x \in \mathbb{R}$:**

Analysis equation:

$$\hat{f}(s) = \int_{x=-\infty}^{\infty} f(x) e^{-2\pi i sx} \, dx \quad (2.2.1)$$

Synthesis equation:

$$f(x) = \int_{s=-\infty}^{\infty} \hat{f}(s) e^{2\pi i sx} \, ds \quad (2.2.2)$$

**Case 2. $x \in \mathbb{T}_p$, where $\mathbb{T}_p$ is a circle of circumference $p$:**

Thus $f$ is a $p$-periodic function on $\mathbb{R}$, $f(x + p) = f(x)$. Clearly we would like the basis functions to be $p$-periodic as well. Therefore

$$e^{2\pi i s(x + p)} = e^{2\pi i sx}$$

$\Rightarrow$ $2\pi sp = 2\pi n$, $n \in \mathbb{Z}$

$\Rightarrow s = \frac{n}{p} \quad (2.2.3)$

The function $f$ or the basis need not be restricted to the interval $[0, p]$. Therefore $n \in \mathbb{Z}$.

Analysis equation:

$$\hat{f}(n) = \frac{1}{p} \int_{x=0}^{p} f(x) e^{-\frac{2\pi in}{p}} \, dx \quad (2.2.4)$$

Synthesis equation:

$$f(x) = \sum_{n=-\infty}^{\infty} \hat{f}(n) e^{\frac{2\pi in}{p}} \quad (2.2.5)$$
Case 3. \( f \) is defined only on \( n \in \mathbb{Z} \):
The basis functions are thus given by \( e^{2\pi i sn} \), \( s \in \mathbb{R} \). Now, since \( n \in \mathbb{Z} \),
\[
e^{2\pi i (s+k)n} = e^{2\pi i sn}, \forall k \in \mathbb{Z}
\]
Now, if we let \( 0 \leq s < p \), \( e^{\frac{2\pi i sn}{p}} \) is \( p \)-periodic in \( s \). Therefore \( s \in \mathbb{T}_p \).
Analysis equation:
\[
\hat{f}(s) = \frac{1}{p} \sum_{n=-\infty}^{\infty} f(n) e^{-\frac{2\pi i sn}{p}}
\]  \( (2.2.6) \)
Synthesis equation:
\[
f(n) = \int_{s=0}^{p} \hat{f}(s) e^{\frac{2\pi i sn}{p}} \, ds
\]  \( (2.2.7) \)

Case 4. The Discrete Fourier Transform:
Let \( \mathbb{P}_N \) denote the set of \( N \) uniformly-spaced points on the circle \( \mathbb{T}_N \). Let \( f \) be defined on \( \mathbb{P}_N \). Thus \( f \) is a \( N \)-periodic function defined on \( \mathbb{Z} \). We would like the basis functions to be \( N \)-periodic, too. Therefore
\[
e^{2\pi i (s+N)n} = e^{2\pi i sn}, n \in \mathbb{Z}
\]
\[
\Rightarrow e^{2\pi i sn} = 1
\]
\[
\Rightarrow sN = m, m \in \mathbb{Z}
\]
\[
\Rightarrow s = \frac{m}{N}
\]  \( (2.2.8) \)
Thus
\[
e^{\frac{2\pi i sn}{N}} = e^{\frac{2\pi i (m+kn)n}{N}} = e^{\frac{2\pi i (m+kn)n}{N}}, k \in \mathbb{Z}.
\]
Thus \( e^{\frac{2\pi i sn}{N}} \) gives us unique values for \( m = \{0, 1, \ldots, N-1\} \), whereafter it repeats itself. The Fourier Transform equations of a function defined on \( \mathbb{P}_N \) is given as Analysis equation:
\[
\hat{f}(m) = \frac{1}{N} \sum_{n=0}^{N-1} f(n) e^{-\frac{2\pi i mn}{N}}
\]  \( (2.2.9) \)
Synthesis equation:
\[
f(n) = \sum_{m=0}^{N-1} \hat{f}(m) e^{\frac{2\pi i mn}{N}}
\]  \( (2.2.10) \)
This is the form of the Fourier Transform that is most often used in image processing owing to the discrete nature of the image data.

Unfortunately, although a Fourier transform would tell about the presence of the frequencies present in the signal, the location where the frequencies occur is unavailable.
2.3 Gabor Transforms

A modification to the Fourier Transform is the Gabor Transform (also called the Short Time Fourier Transform or the Windowed Fourier Transform). In this case, a window function (usually a Gaussian) is translated across the domain of the signal and the Fourier transform is computed on the product of the original signal and the Gaussian. The Gaussian decays exponentially and serves to limit the original signal such that 95% of the original signal is within 3 standard deviations from the center of the Gaussian function. Thus, as the Gaussian function is translated over the domain of the signal, the Fourier transform gives us the frequencies present within that window.

2.4 The Wavelet Transform

Like the Fourier transform and the STFT, in the previous sections, the same central theme of analysis continues here. The wavelet transform also seeks to represent a function in terms of some basis functions and their coefficients. We will see that in the case of the discrete wavelet transform a weaker requirement than that of a basis is sufficient. We will introduce frames which can be viewed as redundant bases. The input function as well as the so-called basis functions have to satisfy certain criteria. Such conditions are not meant to be restrictive. On the contrary they ensure that the transform process is well defined (for example that the integral involved is finite, etc.).

As the name suggests the wavelet is a small wave or part of a wave. Limited support is preferred over a stationary wave, since the motivation behind its use is to represent local behaviour. Not every function can qualify to be a wavelet. The conditions placed on a function for it to qualify it as a wavelet, buy us other very desirable properties as discussed later.

A wavelet $\psi$ is a function that should at least satisfy the following properties:

$$\psi \in L^2$$  \hspace{1cm} (2.4.1)

So $\psi$ is square integrable. We should also have:

$$\int_{\mathbb{R}} \frac{|\hat{\psi}(\eta)|^2}{\eta} d\eta < \infty$$  \hspace{1cm} (2.4.2)

From the study of Fourier transforms we already know the following theorem:

**Theorem 2.4.1** If $g \in L^1$, the its Fourier transform $\hat{g}$ is continuous and

$$\lim_{\eta \to \pm \infty} \hat{g}(\eta) = 0$$

*See [Bla98]*

Condition 2.4.2 above is similar to the result of the above theorem. We instead require that $|\hat{\psi}(\eta)|^2$, the square of the Fourier transform of the wavelet function, go to zero at least faster than $\frac{1}{\eta}$ as $\eta \to \pm \infty$. The two conditions 2.4.1 and 2.4.2 above thus attempt to somewhat localize the wavelet.
function in both the time and frequency domains. Of course, we here do not require that $\psi \in L^1$. However, in practice, all wavelets are also $L^1$ functions as well. Thus $\psi \in L^1 \cap L^2$.

In addition we require that,

$$\| \psi \| = 1$$

(2.4.3)

where the norm is of course the $L^2$ norm. Thus a wavelet can be thought to have unit mass. This is then analogous to the unit vectors of the basis of a vector space.

For a function $\psi \in L^2$ it is not always clear at first glance if it does in fact satisfy condition 2.4.2. But if in addition to condition 2.4.1, $\psi$ also satisfies the condition $t \psi \in L^1$ (i.e., $\psi$ goes to zero at least faster than $1/t$ as $t \to \pm \infty$), then we have the following theorem:

**Theorem 2.4.2** If $\psi \in L^2$ and $t \psi \in L^1$, then

$$\int_{\mathbb{R} \setminus 0} \frac{|\hat{\psi}(\eta)|^2}{\eta} d\eta < \infty \iff \int_{-\infty}^{+\infty} \psi(t) dt = 0$$

The proof is quite straightforward, see [Bla98] sec: 3.1. Note also that

$$\int_{-\infty}^{+\infty} \psi(t) dt = \hat{\psi}(0)$$

Clearly then, a wavelet must have both positive as well as negative values and in equal measure. In other words, the wavelet has zero mean. However the wavelet need not exhibit symmetry. The intuitive reasoning behind this property is that the wavelet should not add anything to the input signal that it seeks to help represent. For a wavelet to be useful several other properties are desired. These include continuity, differentiability, compact support, etc.

### 2.4.1 The Continuous Wavelet Transform

Suppose that $\psi$ is a function that satisfies conditions 2.4.1, 2.4.2 and 2.4.3. We keep $\psi$ fixed and call it the *mother wavelet*. Other wavelet bases will be derived from this *mother wavelet* in the form of dilates and translates of $\psi$.

**Definition 2.4.1 Dilation**

*Given a function $\psi(t)$, $t \in \mathbb{R}$ and some $a \in \mathbb{R}$, $a > 0$,*

$$\psi_a(t) := \frac{1}{|a|^2} \psi\left(\frac{t}{a}\right)$$

*is defined to be the dilation of $\psi$ by a factor of $a$.*

The factor $a$ is known as the dilation or scaling parameter. It essentially stretches the original function by a factor of $a$. For values of $a < 0$, the original function is reflected about the vertical axis and stretched by the same amount. The function $\psi_a(t)$ obtained after dilating $\psi$ is also a wavelet, i.e. it satisfies the three conditions mentioned before. Condition 2.4.1 is trivially satisfied.
Condition 2.4.2 follows from the fact that \( \tilde{\psi}_a(\eta) = (\tilde{\psi})_{\frac{b}{a}}(\eta) \). This can be easily verified. To verify the third condition we see that the \( L^2 \)-norm of \( \psi_a \) is given by

\[
\| \psi_a \| = \int |\psi_a(t)| \, dt
\]

\[
= \frac{1}{|a|} \int \left| \psi(\frac{t}{a}) \right|^2 \, dt
\]

(let \( y = \frac{t}{|a|} \), therefore \( dt = |a| \, dy \))

\[
= \frac{1}{|a|} \int \left| \psi(y) \right|^2 |a| \, dy
\]

\[
= \int \left| \psi(y) \right|^2 \, dy
\]

\[
= 1 \text{ (from condition 2.4.3 since } \psi \text{ is already a wavelet)}.
\]

The scaling parameter is usually restricted to \( \mathbb{R}_{>0} = \{ x \in \mathbb{R} \mid x > 0 \} \).

The other operation that is performed on the wavelet is translation. Thus if \( \psi \) be any wavelet and \( b \in \mathbb{R} \), then \( \psi_b(t) = \psi(t - b) \) is said to be the translation of \( \psi \) by the parameter \( b \). It is easy to see that the translation operation also satisfies the conditions for a wavelet function. Thus, putting dilation and translation together we obtain the expression for a wavelet function

\[
\psi_{a,b}(t) = \frac{1}{|a|^2} \psi\left( \frac{t-b}{a} \right)
\]

that is obtained from a given wavelet \( \psi \) after being dilated by \( a \) and translated by \( b \).

**Definition 2.4.2** *The Continuous Wavelet Transform*

The wavelet transform \( W_{\psi} f : \mathbb{R}^* \times \mathbb{R} \to \mathbb{C} \) of a function \( f \in L^2 \) with respect to the wavelet function \( \psi \) is given by

\[
W_{\psi} f(a,b) := \langle f, \psi_{a,b} \rangle = \frac{1}{|a|^2} \int f(t) \overline{\psi\left( \frac{t-b}{a} \right)} \, dt
\]

where \( \mathbb{R}^* := \mathbb{R} \setminus \{0\}, \ a \in \mathbb{R}^* \) and \( b \in \mathbb{R} \).

As implied by the scalar product representation of wavelet transform, \( W_{\psi} f(a,b) \) is the coefficient of the function \( f \) when projected onto the wavelet function \( \psi_{a,b} \). Thus, the collection \( \{ W_{\psi} f(a,b) \}_{(a,b) \in \mathbb{R}^* \times \mathbb{R}} \) forms the set of coefficients for the corresponding basis functions \( \psi_{a,b} \), such that together they form a representation of the function \( f \). This brings us to the definition of the inverse wavelet transform.

**Theorem 2.4.3** *The Inverse Wavelet Transform*

Under suitable regularity and admissibility conditions for \( f \) and \( \psi \) the inverse wavelet transform is given by

\[
f(t) := \frac{1}{C_{\psi}} \int_{\mathbb{R}^* \times \mathbb{R}} W_{\psi} f(a,b) \psi_{a,b}(t) \, \frac{dadb}{|a|^2}
\]
where \( f \) is continuous at the point \( t \) and \( C_\psi \) is the wavelet constant given by

\[
C_\psi := 2\pi \int_{\mathbb{R}} \frac{|\widehat{\psi}(\eta)|^2}{\eta} d\eta
\]

The regularity of a function \( f \) translates roughly into the smoothness of the function. One way to quantify the smoothness of a function is by the number of times that it can be continuously differentiated. The smoother the function \( f \), the faster is the decay of its Fourier transform \( \widehat{f}(\eta) \) as \( \eta \to \pm \infty \). It is also useful to note the dual of the previous statement: the faster that \( f(x) \to 0 \) as \( x \to \pm \infty \), the smoother is its Fourier transform \( \widehat{f} \).

An example of a space of smooth functions is the Schwartz space \( S \), which is the space of all functions \( f : \mathbb{R} \to \mathbb{C} \) with the following properties:

- \( f \in C^\infty(\mathbb{R}) \)
- \( f^{(n)}(x) \to 0 \) faster than \( \frac{1}{|\eta|^r} \), \( \forall \ n, r \in \mathbb{Z}_+ \), where \( f^{(n)} \) represents \( f \) differentiated \( n \) times.

It is easily seen that \( S \subseteq L^1(\mathbb{R}) \cap L^2(\mathbb{R}) \). In fact \( S \subseteq L^p(\mathbb{R}) \), \( \forall \ p \geq 1 \). Note also that \( f \) need not be continuous everywhere. Piecewise continuity is fine.

The admissibility condition is the same as condition 2.4.2 described previously. In other words we require \( 0 < C_\psi < \infty \). Also the conditions for the wavelet transform in Definition 2.4.2 also apply for the inverse wavelet transform since the wavelet coefficients \( \{ W_\psi f(a, b) \} \) come from the forward transform.

Finally, we note that the measure used in the integral for the inverse wavelet transform is \( d\mu = \frac{da \, db}{|a|^2} \) and not \( d\mu = da \, db \) as one would imagine at first. Intuitively, this is because the effects of \( a \) and \( b \) are not the same. \( \frac{da \, db}{|a|^2} \) represents the left invariant Haar measure on the following locally compact Hausdorff Topological group of affine transforms:

\[
G = \{ T_{a,b} \mid a \in \mathbb{R}^* \text{ and } b \in \mathbb{R} \}
\]

where

\[
T_{a,b} : \mathbb{R} \to \mathbb{R} \text{ such that } t \to at + b , \text{ is an affine transform.}
\]

The group operation here is composition. Thus the scaling and translation operations on the wavelet function \( \psi \) can be represented as

\[
\psi_{a,b}(t) = \frac{1}{|a|^2} \psi \left( \frac{t-b}{a} \right) = \frac{1}{|a|^2} \psi \left( T_{a,b}^{-1} t \right)
\]

\( T_{a,b}^{-1} \) also belongs to the group \( G \). Thus for a left invariant measure on \( \mathbb{R}^* \times \mathbb{R} \), the group \( G \) of affine transforms gives us a useful parameterization of that space. \( (a, b) \in \mathbb{R}^* \times \mathbb{R} \) is represented
by $T_{a,b} \in G$. Thus a measure $\mu$ on $\mathbb{R}^* \times \mathbb{R}$ is said to be left invariant if for a function $\psi$ with domain $\mathbb{R}^* \times \mathbb{R}$, we have

$$
\int_G \psi(T_{s,t}T_{a,b}) \, d\mu = \int_G \psi(T_{a,b}) \, d\mu
$$

(2.4.4)

Now $T_{s,t}T_{a,b}x = T_{s,t}(ax + b) = sax + sb + t = T_{sa, sb + t}$. Let $d\mu = \frac{da\, db}{a^2}$. Thus we have

$$
\int_G \psi(T_{s,t}T_{a,b}) \, d\mu = \int_{\mathbb{R}^* \times \mathbb{R}} \psi(sa, sb + t) \frac{da\, db}{a^2}
$$

(2.4.5)

Let $u = sa$. Therefore $du = sda$ and $a = \frac{u}{s}$. Let $v = sb + t$. Therefore $dv = sdb$. Replacing these values into equation 2.4.5 we have

$$
\int_{\mathbb{R}^* \times \mathbb{R}} \psi(sa, sb + t) \frac{du\, dv}{s^2 \, v^2} = \int_{\mathbb{R}^* \times \mathbb{R}} \psi(u, v) \, du\, dv
$$

$$
= \int \psi(u, v) \, du\, dv
$$

$$
= \int_G \psi(T_{a,b}x) \, d\mu
$$

Hence, the selected measure $d\mu = \frac{da\, db}{a^2}$ is left invariant.

The domain of the continuous wavelet transform is uncountable. Hence it is computationally prohibitive to try and obtain the values for the function $W_\psi f$ at every $(a, b) \in \mathbb{R}^* \times \mathbb{R}$. It would be useful if it were possible to represent $f$ using only a finite number of wavelet coefficients. The discrete wavelet transform helps us do just that.

### 2.4.2 The Discrete Wavelet Transform

The continuous wavelet transform helps us to represent a function $f(t) \in \mathbb{R}$ as a function $W_\psi f$ of two variables $a \in \mathbb{R}^*$ and $b \in \mathbb{R}$. This introduces a lot of redundancy into the representation. By taking advantage of this redundancy we are able to reconstruct the original function $f$ from a sampling of the coefficients $\{W_\psi f(a, b)\}$. Note that we do not require $f$ to be bandlimited.

**Definition 2.4.3** The Discrete Wavelet Transform

The discrete wavelet transform is the wavelet transform evaluated at only a discrete set of points $(a, b) \in \mathbb{R}_{>0} \times \mathbb{Z}$.

We thus seek to obtain a discrete (countable or preferably finite) set of coefficients

$$
c_{m,n} = W_\psi f(a_m, b_m, n), \quad (m, n) \in \mathbb{Z} \times \mathbb{Z}
$$

with which to represent the function $f$. By representation we mean that the function $f$ should be reconstructable from the coefficients $c_{m,n}$.

The discrete set of points where $W_\psi f$ is sampled is based upon a grid which is characterized by a variable $\sigma$ called the zoom step. This value usually remains constant for the transform process and
is usually set to $\sigma = 2$ (also known as the dyadic step). The discrete sample points $a_m$ and $b_m, n$ are now determined by the expressions

$$a_m = \sigma^m$$

$$b_{m,n} = \beta \sigma^m n$$

where $m, n \in \mathbb{Z}$. There are corresponding results for $a$ and $b$, but we will restrict our discussion to the grid values described above. The constant $\beta > 0$ is called the base step. Usually its value is set to $\beta = 1$.

Thus the $\mathbb{R}_{>0} \times \mathbb{R}$ half-plane is divided into a grid of rectangles. Each point $(a_m, b_m, n)$ can be thought of as representing a rectangle $R_{m,n} = \left[ a_{m, n - \frac{1}{2}}, a_{m, n + \frac{1}{2}} \right] \times \left[ b_{m, n - \frac{1}{2}}, b_{m, n + \frac{1}{2}} \right]$. Thus the measure content for the point $(a_m, b_m, n)$ is given by the $\mu$-content of the enclosing rectangle

$$\mu(R_{m,n}) = \int_{\sigma^m - \frac{1}{2}}^{\sigma^m + \frac{1}{2}} \frac{dadb}{a^2}$$

$$= \beta \sigma^m \left( n + \frac{1}{2} - (n - \frac{1}{2}) \right) \int_{\sigma^m - \frac{1}{2}}^{\sigma^m + \frac{1}{2}} \frac{da}{a^2}$$

$$= \beta \sigma^m \cdot 1 \cdot \left( \frac{1}{\sigma^{m + \frac{1}{2}}} - \frac{1}{\sigma^{m - \frac{1}{2}}} \right)$$

$$= \beta \left( \frac{1}{\sigma^{\frac{1}{2}}} - \frac{1}{\sigma^{\frac{1}{2}}} \right)$$

$$= \frac{\beta (\sigma - 1)}{\sigma^2}$$

and is therefore independent of the position $(m, n)$ within the grid. This is a direct benefit of selecting an invariant Haar measure as discussed earlier.

Let $\psi_{m,n} = \psi_{a_m, b_m, n}$. We want to retain only a few of the wavelet functions from the original full set $\psi_{a,b}$, $(a, b) \in \mathbb{R}^* \times \mathbb{R}$. It remains to be seen if the above method of sampling yields a basis or at least a frame that may be used to represent the function $f$.

### 2.4.3 Bases and Frames

In this section we will restrict our discussion to finite dimensional Hilbert spaces. The same concepts apply to the infinite-dimensional case as well. Let $X$ be an $k$-dimensional Hilbert space. A basis for $X$ is a collection $\{\rho_n\}_{n=1}^k$ of elements in $X$ such that

1. $\{\rho_n\}_{n=1}^k$ spans $X$, i.e. any element $x \in X$ can be written as a linear combination of the basis elements. In other words there exists a collection $\{c_n\}_{n=1}^k, (c_n \in \mathbb{R})$, of coefficients such that $x = \sum_{n=1}^k c_n \rho_n$.

2. The $\{\rho_n\}_{n=1}^k$ are linearly independent, i.e. the combination $x = \sum_{n=1}^k c_n \rho_n$ in (1) above is unique.
Note that orthogonality in not implied although it is usually a desired property. Given an element $x \in X$, an orthonormal basis $\{\rho_n\}_{n=1}^k$ gives us a unique set of coefficients $c_n = \langle x, \rho_n \rangle$ with which to represent each such element $x$. Let $B_\rho : X \rightarrow C^k$ be the map defined as $B_\rho x = \sum_{n=1}^k \langle x, \rho_n \rangle e_n$ where $(e_1, e_2, \cdots, e_n)$ is the canonical basis for the space $C^k$. Thus the $n$-th coefficient of $x$ with respect to the basis is given by $c_n = B_\rho x(n) = \langle B_\rho x, e_n \rangle = \langle x, \rho_n \rangle$. Now, if we assume that $\langle x, \rho_n \rangle = 0$ for all $n \in \mathbb{N}$, then $x \perp \rho_n$ for all $n \in \mathbb{N}$. Thus $x \perp \text{span}(\rho_1, \rho_2, \cdots, \rho_n)$ and now since $\text{span}(\rho_1, \rho_2, \cdots, \rho_n) = X$, it follows that $x = 0$. So since $\{\rho_n\}_{n=1}^k$ is a basis, $B_\rho$ is an injective mapping i.e. it maps distinct elements in $X$ to distinct elements in $C^k$. Also, for a basis $\{\rho_n\}_{n=1}^k$ we have

$$\|x\|_{L^2}^2 = \langle x, x \rangle$$

$$= \langle x, \sum_{n=1}^k c_n \rho_n \rangle$$

$$= \sum_{n=1}^k c_n \langle x, \rho_n \rangle$$

$$= \sum_{n=1}^k \|c_n\|^2$$

$$= \|B_\rho x\|_{L^2}^2$$

For a basis, the map $B_\rho$ is always isometric. Although a basis is very convenient from a unique representation point of view, the conditions for its existence, i.e. linear independence and the fact that it has to span the space, are restrictive when one wishes to add further conditions on the basis. For this reason we consider Riesz bases. Before introducing the concept of a Riesz basis, we introduce the related concept of a frame (only finite dimensional Hilbert spaces are considered in this discussion).

A frame can be thought of as a redundant or over-complete basis. The second property, that of linear independence, is not required. Let $\{\rho_n\}_{n=1}^r$ be a set of elements in $X$ (recall that $k$ is the dimension of the finite dimensional Hilbert space $X$), such that $\text{span}\{\rho_n\}_{n=1}^r = X$.

Let $T_\rho : X \rightarrow C^r$ be a map such that

$$T_\rho x := \langle x, \rho_n \rangle e_n$$

where again $(e_1, e_2, \cdots, e_r)$ is the canonical basis for $C^r$. The $n$th co-ordinate of $T_\rho x$ is given by $c_n = \langle T_\rho x, e_n \rangle = \langle x, \rho_n \rangle$. $T_\rho$ is known as the frame operator for the frame $\{\rho_n\}_{n=1}^r$.

Since the elements of frame are not necessarily linearly independent, $T_\rho$ is essentially a one-many map. Due to the redundancy introduced by $\{\rho_n\}_{n=1}^r$, an element of $X$ can be represented by more than one co-ordinate set in $C^r$. An injective map is very desirable, since then we can work towards defining a bijective inverse map $T_\rho^{-1} : \text{im}(T_\rho) \rightarrow X$ , where $\text{im}(T_\rho) = \{T_\rho x \mid x \in X\}$. The space $C^r$ equipped with a scalar product definition

$$\langle a, b \rangle = \sum_{n=1}^r a_n \overline{b_n} \quad \forall \ a, b \in C^r$$

14
is also a Hilbert space.

**Definition 2.4.4 Frame**

Let $X$ be a $k$-dimensional Hilbert space ($k$ being a finite positive integer). A collection $\{\rho_n\}_{n=1}^r$ of elements in $X$ is a frame for $X$ if and only if there exists constants (called frame constants or frame bounds) $A, B \in \mathbb{R}$ and a frame operator $T_\rho : X \to \mathbb{C}^r$ such that

1. $0 < A \leq B$
2. $A\|x\|^2 \leq \|T_\rho x\|_2^2 \leq B\|x\|^2 \forall x \in X.$

The second condition can also be viewed in the scalar product form:

$$A\langle x, x \rangle_X \leq \langle T_\rho x, T_\rho x \rangle_{\mathbb{C}^r} \leq B\langle x, x \rangle_X \forall x \in X.$$

If $A = B$, the frame is called a tight frame and clearly

$$A\|x\|_2^2 = \|T_\rho x\|_2^2 \quad (2.4.6)$$

**Definition 2.4.5 Adjoint of a Frame Operator**

Let $X$ and $Y$ be Hilbert spaces. The adjoint $T^*$ of a mapping $T : X \to Y$ is a mapping $T^* : Y \to X$ such that

$$\langle x, T^* y \rangle_X = \langle T x, y \rangle_Y \forall x \in X, \forall y \in Y.$$

**Definition 2.4.6 Dual Frame**

Let $X$ be a $k$-dimensional Hilbert space and $\{\rho_n\}_{n=1}^r$ be a frame for $X$ with respect to the frame operator $T_\rho$. The collection of elements $\tilde{\rho} := \{\tilde{\rho}_n\}_{n=1}^r$ such that

$$\tilde{\rho}_n = \left( T_\rho^* T_\rho \right)^{-1} \rho_n$$

is known as the dual frame for the frame $\rho$.

$\tilde{\rho}$ is a frame with frame constants $\frac{1}{A} \geq \frac{1}{B} > 0$ where $A$ and $B$ are the frame constants for the frame $\rho$ as defined previously. Dual frames let us define an expression for the reconstruction of an element $x \in X$ as follows

$$x = \sum_{n=1}^r \langle x, \rho_n \rangle \tilde{\rho}_n = \sum_{n=1}^r \langle x, \rho_n \rangle \rho_n$$

The corresponding frame operator for the dual frame $\tilde{\rho}$ is given by

$$\tilde{T}_\rho = T_\rho \left( T_\rho^* T_\rho \right)^{-1} = \left( T_\rho^* \right)^{-1}$$

Although frames form a very useful part of wavelet theory, they are not used in the main application of wavelet theory for this thesis, i.e. multiresolution analysis. As discussed in a later section, we will be concerned only with orthonormal bases. This brings us back to the discussion on bases and the definition of the Riesz basis.
Definition 2.4.7 Riesz Basis
Let $X$ be a separable complex Hilbert space and $\Gamma$ be a countable index set. Then the sequence of elements $\{\rho_i\}_{i \in \Gamma} \subset X$ is a Riesz basis if

1. $\{\rho_i\}_{i \in \Gamma}$ is a complete (total) sequence, i.e. $\text{span} \left( \{\rho_i\}_{i \in \Gamma} \right) = X$.
2. There exist constants $0 < A \leq B < \infty$ such that for any sequence $\{c_i\}_{i \in \Gamma} \in l^2(\Gamma)$ we have

\[
A \sum_{i \in \Gamma} |c_i|^2 \leq \left\| \sum_{i \in \Gamma} c_i \rho_i \right\|^2 \leq B \sum_{i \in \Gamma} |c_i|^2
\]

The left inequality in condition (2) above clearly implies linear independence. Constants $A$ and $B$ are known as the lower and upper Riesz bounds respectively. The supremum over all the lower Riesz bounds is called the optimal lower Riesz bound. Similarly the infimum over all the upper Riesz bounds is called the optimal upper Riesz bound. A Riesz basis, or a frame for that matter, is called tight if $A = B$. The bounds $A$ and $B$ represent the redundancy of the Riesz basis. The constant $A$ is the minimum redundancy factor. If $A > 1$ then the Riesz basis is redundant. A Riesz basis (or a frame) is also an orthonormal basis if and only if it is tight with $A = B = 1$. Let $\{\rho_i\}_{i \in \Gamma}$ be a tight Riesz basis such that $A = B \neq 1$. Now, let $\tilde{\rho}_i := \frac{\rho_i}{\sqrt{A}}$. Then $\{\tilde{\rho}_i\}_{i \in \Gamma}$ is clearly an orthonormal basis. Thus any tight Riesz basis can be easily converted to an orthonormal basis and that is a very convenient result.

Theorem 2.4.4 Let $\rho := \{\rho_m\}_{m \in M}$ be a Riesz basis, where $M$ is a countable index set. Let $0 < A \leq B$ be the Riesz constants of the basis. Then $\rho$ is also a frame, with frame constants $A$ and $B$.

A Riesz basis can be thought of as something between a basis and a frame. It is infact a countable frame whose elements are linearly independent.

2.4.4 Multiresolution

The previous sections have discussed the mother wavelet $\psi$ and its dilates and translates. Multiresolution seeks to represent a given function $f \in L^2$ at various resolutions. The discrete version of the wavelet transform involves a zoom step $\sigma$ which gives us the different levels of dilation. What we now discuss will be in the realm of the discrete wavelet transform, since the resolutions are generally assumed to be integer powers of a zoom step as detailed previously in the section on the discrete wavelet transform. As mentioned before, for a function $f \in L^2$, the discrete wavelet transform coefficients at step $(m,n) \in \mathbb{Z} \times \mathbb{Z}$ is given by

\[
W_{\psi}f[m,n] := W_{\psi}f(a_m,b_{m,n}) = \frac{1}{\sigma^m} \int_{-\infty}^{+\infty} f(t) \psi \left( \frac{t - \beta m}{\sigma^m} \right) dt = \langle f, \psi^{m,n} \rangle
\]
where, as before, $a_m = \sigma^m$ and $b_{m,n} = \beta \sigma^m n$ and $\sigma > 0$ is the zoom step which is usually set to 2. The wavelet coefficient can be viewed as the coordinates of a projection $P_m : L^2 \to \mathbb{C}^N$, where $N \in \mathbb{Z}$, of the function $f$ into the space spanned by all the translates of the mother wavelet $\psi$ dilated to level $m$. Thus each wavelet coefficient at the $(m,n)$ step can be viewed as the scalar product of $f$ with a dilated and translated version of the mother wavelet $\psi$. Similarly the wavelet coefficients at level $(m+1)$ are given by:

$$
W_\psi f [m+1, n] := W_\psi f (a_{m+1}, b_{m+1}, n) \\
= \frac{1}{\sigma^{m+1}} \int_{-\infty}^{+\infty} f(t) \psi \left( \frac{t - \beta \sigma^{m+1} n}{\sigma^{m+1}} \right) dt \\
= \left< f, \psi_{\sigma^{m+1}, \beta \sigma^{m+1} n} \right>
$$

To obtain the wavelet coefficients at level $m+1$, we would have to compute every coefficient as per the discrete wavelet transform equation mentioned above. With no clear link, as yet, between the wavelet functions at the $m$ and $m+1$ levels, one would have to compute the integral on the right-hand side of the equations above, for each and every coefficient corresponding to the grid point $(m+1, n)$. We would have to follow the same procedure for the coefficients at all other levels. Clearly, this procedure becomes computationally prohibitive very fast.

The main idea behind dilation is the representation of a function $f \in L^2$ using the mother wavelet function $\psi$ at different scales (see figure 2.1). Let us use the following notation:

$$
\psi_{[m,n]} := \psi_{a_m, b_{m,n}}
$$

Higher levels increase the support of $\psi_{[m,n]}$ and lower levels decrease it. The number of elements in the frame that is used to represent $f$, is higher at lower dilation levels. On the other hand, the main motivation behind translation is to be able to cover the whole time axis ($t \in \mathbb{R}$) with copies of the dilated wavelet. The translation factor is an integer multiple of $\beta \sigma^m$. So at higher levels, the step size is larger. This is most convenient since orthogonality is a much desired feature for a basis, and so we wish to minimize any unnecessary overlap between the any two consecutive translates of the wavelet function at the same level. So when the size of the support changes, we would like the translation step size to be compensated accordingly.

The main idea in multiresolution is to be able to represent the wavelet function using a scaled version of itself. We wish to look at the same function $f$ at different scales and so it would be good to have an inter-relationship between the dilation levels. Ideally, for $s > r (s, r \in \mathbb{Z})$, we would like to be able to represent $\psi_{[s,n]}$ using some of the functions from $\psi_{[r,\cdot]}$ in a simple linear summation form like:

$$
\psi_{[s,n]}(t) = \sum_{k=0}^{N} c_k \psi_{[r,k+n]}(t)
$$

where $N \in \mathbb{Z}$ and is preferably finite.
This has the immediate benefit of reducing the computation time, since now we have

\[
W_\psi f [s, n] = \int_{\mathbb{R}} f(t) \psi_{[s,n]}(t) dt
\]

\[
= \int_{\mathbb{R}} f(t) \left[ \sum_{k=0}^{N} c_k \psi_{[s,n+k]}(t) \right] dt
\]

\[
= \sum_{k=0}^{N} c_k \int_{\mathbb{R}} f(t) \psi_{[s,n+k]}(t) dt
\]

\[
= \sum_{k=0}^{N} c_k W_\psi f [r, n + k]
\]

So if we have already computed the coefficients for level \( r \), i.e. \( \{W_\psi f[r, \cdot]\} \), then obtaining the coefficients for a higher level \( s \) reduces to computing a simple sum. It requires much less computation than having to compute an integral over all of \( \mathbb{R} \) for each and every coefficient. In essence, multiresolution is a construction which enables us to represent a function at a higher (coarser) level without having to recompute the integrals. Henceforth, in our discussion of multiresolution analysis, we will assume \( \sigma = 2 \) and \( \beta = 1 \). Also, we choose \( \psi \) such that the \( \psi_{[m,n]} \) form an orthonormal basis of \( L^2(\mathbb{R}) \). Then we do not have to use dual frames to represent the original function \( f \) using the wavelet coefficients.

As specified in the definition, the wavelet transform operates on functions in \( L^2(\mathbb{R}) \). For multiresolution analysis, we divide \( L^2(\mathbb{R}) \) into a sequence of spaces \( \{V_i\}_{i \in \mathbb{Z}} \) with special conditions.

**Definition 2.4.8 Multiresolution**

Multiresolution is a construction of a sequence of closed subspaces of \( L^2(\mathbb{R}) \) such that the following conditions are satisfied:
1) **Inclusions:** \( \cdots \subset V_{i+1} \subset V_i \subset \cdots \subset V_{-1} \subset \cdots \subset V_{-k} \subset \subset L^2(\mathbb{R}) \).

2) **Completeness:** 
   \[
   \bigcup_{i=-\infty}^{+\infty} V_i = \lim_{i \to -\infty} V_i = L^2(\mathbb{R})
   \]

3) **Separation:** 
   \[
   \bigcap_{i=-\infty}^{+\infty} V_i = \lim_{i \to +\infty} V_i = \{0\}, \text{ the zero function.}
   \]

4) **Translations:** 
   \[f(t) \in V_i \iff f(t-2^i n) \in V_i.\]

5) **Dilations:** 
   \[f\left(\frac{t}{2^n}\right) \in V_{j+i} \iff f\left(\frac{t}{2^n}\right) \in V_i.\]

6) **Basis:** \( \exists \text{ a function } \phi \in L^1(\mathbb{R}) \text{ such that } \{\phi(t-n)\}_{n \in \mathbb{Z}} \text{ is a Riesz basis for } V_0. \) In fact, it is desirable to have a function \( \phi \in L^1 \cap L^2 \text{ such that } \{\phi(t-n)\}_{n \in \mathbb{Z}} \text{ form an orthonormal basis for } V_0. \)

**Theorem 2.4.5** Let \( \{V_i\}_{i \in \mathbb{Z}} \) be a multiresolution approximation for \( L^2(\mathbb{R}) \) such that \( \{\theta(t-n)\}_{n \in \mathbb{Z}} \) is the corresponding Riesz basis. Let \( \phi \) be a function such that its Fourier transform is given by 
\[
\hat{\phi}(\eta) = \frac{\hat{\theta}(\eta)}{\left(\sum_{n=-\infty}^{+\infty} \left|\hat{\theta}(\eta + 2\pi n)\right|^2\right)^{\frac{1}{2}}}
\]
and define 
\[
\phi_{[i,n]}(t) := \frac{1}{2^i} \phi\left(\frac{t-2^i n}{2^i}\right)
\]
where \( i, n \in \mathbb{Z}. \) Then the collection \( \{\phi_{[i,n]}\}_{n \in \mathbb{Z}} \) of functions is an orthonormal basis of the space \( V_i. \)

From the above definition of multiresolution, we can describe the **base space** \( V_0 \subset L^2(\mathbb{R}) \) as 
\[
V_0 = \left\{ f \in L^2(\mathbb{R}) \mid f(t) = \sum_{n \in \mathbb{Z}} c_n \phi(t-n) \text{ and } \{c_n\}_{n \in \mathbb{Z}} \in l^2(\mathbb{Z}) \right\}
\]

The approximation of \( f \) is similar to that which might be done using simple functions. Let \( P_i : L^2(\mathbb{R}) \to V_i \) denote the projection operator 
\[
P_i f = \sum_{k=-\infty}^{+\infty} \langle f, \phi_{[i,k]} \rangle \phi_{[i,k]}
\]

Note that \( \{\phi_{[i,n]}\}_{n \in \mathbb{Z}} \) is however not an orthonormal basis for all of \( L^2(\mathbb{R}). \) Now by construction we have \( V_{i+1} \subset V_i \) which implies that \( V_i \setminus V_{i+1} = \emptyset \) and so although \( \{\phi_{[i+1,n]}\}_{n \in \mathbb{Z}} \) is an orthonormal basis for \( V_{i+1}, \) it does not span all of \( V_i. \)

Corresponding to the sequence of spaces \( \{V_i\}_{i \in \mathbb{Z}} \) we can now define another sequence of spaces \( \{W_i\}_{i \in \mathbb{Z}} \) which are known as the **orthogonal complements** of the \( \{V_i\}_{i \in \mathbb{Z}} \) such that 
\[
V_i \perp W_i
\]
and 
\[
V_i = V_{i+1} \oplus W_{i+1}
\]
\[ \forall i \in \mathbb{Z}. \text{ Here } \oplus \text{ denotes a direct sum and } \perp \text{ denotes orthogonality. In other words } \forall f \in V_{i+1} \text{ and } \forall g \in W_{i+1}, f + g = h \in V_i \text{ and is unique. Also } \forall h \in V_i \text{ there exists a unique function } f \in V_{i+1} \text{ and another unique function } g \in W_{i+1} \text{ such that } h = f + g. \text{ Also, } \forall f \in V_{i+1} \text{ and } \forall g \in W_{i+1} \text{ we have } \langle f, g \rangle = 0. \text{ Since both } V_i \text{ and } V_{i+1} \text{ have orthonormal bases, it follows that } W_{i+1} \text{ also has an orthonormal basis. We denote the orthonormal basis for } W_{i+1} \text{ by } \{\psi_{[i+1,n]}\}_{n \in \mathbb{Z}} \text{ and these are in fact the wavelet functions. Clearly then, } \{\phi_{[i+1,n]}\}_{n \in \mathbb{Z}}, \{\psi_{[i+1,n]}\}_{n \in \mathbb{Z}} \text{ together span } V_i. \text{ We know that } \{0\} \in V_i \forall i \in \mathbb{Z}. \text{ It follows that } W_{i+1} \subset V_i. \text{ Note however that } W_{i+1} \neq V_i \setminus V_{i+1}. \]

**Figure 2.2: Multiresolution Spaces**

**Theorem 2.4.6** Let \( \{V_i\}_{i \in \mathbb{Z}} \) denote a multiresolution construction. Let \( \{W_i\}_{i \in \mathbb{Z}} \) be the corresponding orthogonal complements. Then

\[ W_i \perp W_j \forall i \neq j; i, j \in \mathbb{Z}. \text{ Also} \]

\[ \bigoplus_{i=-\infty}^{\infty} W_i = L^2(\mathbb{R}) \]

Let \( Q_i \) denote the orthogonal projection mapping for any \( L^2(\mathbb{R}) \) function onto the space \( W_i \). Then \( Q_{i+1} = P_i - P_{i+1} \). The function \( \phi = \phi_{[0,0]} \) whose integer translates form the orthonormal basis of the space \( V_0 \) for the multiresolution analysis, is known as the scaling function or Father Wavelet. The choice of \( \phi \) is governed by the following criteria:

1) \( \phi \in L^1 \cap L^2 \)
2) \( \int_{\mathbb{R}} \phi(t)dt = 1 \)
3) \( \{\phi_{[0,n]}\}_{n \in \mathbb{Z}} \) are orthonormal.
4) \( \{\phi_{[0,n]}\}_{n \in \mathbb{Z}} \) is a basis for \( V_0 \).
5) \( \phi \) has to be chosen so as to ensure that the inclusion property for the multiresolution analysis is satisfied.

**Theorem 2.4.7** The inclusion property \( V_0 \subset V_{-1} \) of a multiresolution construction holds if and only if \( \phi \) can be written in the following form

\[ \phi(t) = \sqrt{2} \sum_{n=-\infty}^{\infty} h_n \phi \left( \frac{t - 2^{-1}n}{2^{-1}} \right) \text{ a.e. } \forall t \in \mathbb{R} \]

where \( h_n \in \mathbb{C}, \forall n \in \mathbb{Z} \) and \( \{h_n\}_{n \in \mathbb{Z}} \in l^2(\mathbb{Z}) \).
Under some weak conditions, the \( \{h_n\}_{n \in \mathbb{Z}} \) determine the scaling function \( \phi \) uniquely. \( \{h_n\}_{n \in \mathbb{Z}} \) has to be chosen such that integer translates of \( \phi \) are orthogonal and are a basis for \( V_0 \). This can be achieved by ensuring the following relation

\[
\sum_{n=-\infty}^{+\infty} h_n \overline{h_{n+2p}} = \delta_{0,p} \, \forall p \in \mathbb{Z}
\]

It follows from the fact that \( \sum_{n=-\infty}^{+\infty} h_n \overline{h_{n+2p}} = \langle \phi_{[0,p]}, \phi \rangle \). Clearly then

\[
\| \{h_n\}_{n \in \mathbb{Z}} \|_2 = 1 \tag{2.4.7}
\]

The following two results about \( \{h_n\}_{n \in \mathbb{Z}} \) are also useful.

**Theorem 2.4.8** If in addition to the above discussion, \( \{h_n\}_{n \in \mathbb{Z}} \in l^1(\mathbb{Z}) \) then \( \sum_{n=-\infty}^{+\infty} h_n = \sqrt{2} \).

**Theorem 2.4.9** \( \phi \) has compact support \( \Longleftrightarrow \) only finitely many of the \( h_n \) are non-zero.

The values for \( \{h_n\}_{n \in \mathbb{Z}} \) are conveniently obtained in the Fourier domain. The Fourier Transform of the scaling function

\[
\phi(t) := \sqrt{2} \sum_k h_k \phi(2t - k) \quad \text{(for almost all } t \in \mathbb{R})
\]

is given by

\[
\hat{\phi}(\eta) = \frac{1}{\sqrt{2}} \sum_k h_k e^{-ik\eta} \hat{\phi}\left(\frac{\eta}{2}\right)
\]

Let \( H(\eta) := \frac{1}{\sqrt{2}} \sum_k h_k e^{-ik\eta} \). The summation is convergent since \( \| \{h_n\}_{n \in \mathbb{Z}} \|_2 = 1 \). \( H(\eta) \) is called the generating function for the multiresolution construction. Therefore we have:

\[
\hat{\phi}(\eta) = H\left(\frac{\eta}{2}\right) \hat{\phi}\left(\frac{\eta}{2}\right)
\]

**Theorem 2.4.10** The generating function \( H \) for a multiresolution construction has to satisfy the identity

\[
|H(\eta)|^2 + |H(\eta + \pi)|^2 = 1 \quad \text{(for almost all } \eta \in \mathbb{R})
\]

Now, for the space \( W_0 \), we have the following:

**Theorem 2.4.11** \( \psi \in L^2 \) belongs to the space \( W_0 \) if and only if there exists a function \( \gamma \in L^2(\mathbb{R}/2\pi) \) such that \( \hat{\psi} \) can be written as

\[
\hat{\psi}(\eta) := e^{i\eta/2} \gamma(\eta) H\left(\frac{\eta}{2} + \pi\right) \hat{\phi}\left(\frac{\eta}{2}\right)
\]

**Theorem 2.4.12** If we define the mother wavelet \( \psi \) by its Fourier transform as

\[
\hat{\psi}(\eta) := e^{i\eta/2} H\left(\frac{\eta}{2} + \pi\right) \hat{\phi}\left(\frac{\eta}{2}\right) \tag{2.4.8}
\]

then the sequence \( \{\psi_{[0,k]}\}_{k \in \mathbb{Z}} \) is an orthonormal basis for the space \( W_0 \).
Now,
\[
e^{m}H \left( \frac{\eta}{2} + \pi \right) = \frac{1}{\sqrt{2}} \sum_{k} h_{k} e^{i(m\eta + 2\pi i)} e^{m} = \frac{1}{\sqrt{2}} \sum_{k} (-1)^{k} \overline{h_{k}e^{i(m\eta + 2\pi)}} = \frac{1}{\sqrt{2}} \sum_{p} (-1)^{(p-1)} \overline{h_{-(p+1)}e^{i\pi p}}
\]
where \( p = -(k+1) \). Substituting this result into equation 2.4.8 we have
\[
\hat{\Psi}(\eta) = \frac{1}{\sqrt{2}} \sum_{p} (-1)^{(p-1)} \overline{h_{-(p+1)}e^{i\pi p}} \phi \left( \frac{\eta}{2} \right)
\]
which is the Fourier transform of \( \psi(t) = \sqrt{2} \sum_{p} (-1)^{(p-1)} \overline{h_{-(p+1)}} \phi(2t - p) \)

Let \( g_{k} := (-1)^{(k-1)} \overline{h_{-(k+1)}} \). Then we have a representation for the mother wavelet that is similar to that of the scaling function (father wavelet).
\[
\psi(t) = \sqrt{2} \sum_{k} g_{k} \phi(2t - k)
\]

If \( \phi \) has compact support, then using theorem 2.4.9, only finitely many of the \( h_{k} \) are non-zero. If \( \{h_{k}\}_{0 \leq k \leq 2N-1} \) are the only non-zero elements in the corresponding representation for \( \phi \), then the \( g_{k} \) can be represented as
\[
g_{k} := (-1)^{(k-1)} \overline{h_{2N-k-1}}
\]

**Theorem 2.4.13** Assume a multiresolution construction \( \{V_{i}\}_{i \in \mathbb{Z}} \) with scaling function \( \phi \), generating function \( H \) and the mother wavelet defined as
\[
\psi(t) = \sqrt{2} \sum_{k} g_{k} \phi(2t - k)
\]
where \( g_{k} := (-1)^{(k-1)} \overline{h_{-(k+1)}} \) as previously discussed. Then the sequence \( \{\psi_{[i,k]}\}_{[i,k] \in \mathbb{Z} \times \mathbb{Z}} \) defined by
\[
\psi_{[i,k]}(t) = \frac{1}{2^{i}} \psi \left( \frac{t - k2^{i}}{2^{i}} \right)
\]
is an orthonormal basis for \( L^{2}(\mathbb{R}) \).

The following section discusses gives a brief description of the Haar scaling and wavelet functions. The Haar wavelet is the wavelet that has been predominantly tested with the algorithms presented in this thesis.
2.4.5 The Haar Wavelet

The Haar mother wavelet function $\psi$ is defined as

$$\psi_{\text{Haar}}(t) = \psi_{[0,0]}(t) := \begin{cases} 
1, & 0 \leq t < \frac{1}{2} \\
-1, & \frac{1}{2} \leq t < 1 \\
0, & \text{otherwise}
\end{cases}$$

$$= 1_{[0,\frac{1}{2})} - 1_{[\frac{1}{2}, 1]}$$

where $1_A$ represents the indicator function over the set $A \subset \mathbb{R}$ (see figure 2.3). Clearly,

$$\int_{-\infty}^{\infty} \psi(t) \, dt = 0$$

and

$$\int_{-\infty}^{\infty} |\psi(t)|^2 \, dt = 1$$

As before, we define

$${\psi}_{[m,n]} := \frac{1}{2^m} \psi_{\text{Haar}} \left( \frac{t - 2^m n}{2^m} \right)$$

where $(m,n) \in \mathbb{Z} \times \mathbb{Z}$. $\psi_{[m,n]}$ has support on the interval $[n2^m, (n+1)2^m)$. The normalizing factor $\frac{1}{2^m}$ ensures that $\|{\psi}_{[m,n]}\| = 1 \forall (m,n) \in \mathbb{Z} \times \mathbb{Z}$. From theorem 2.4.13, $\{\psi_{[m,n]}\}_{(m,n) \in \mathbb{Z} \times \mathbb{Z}}$ is an orthonormal basis for $L^2(\mathbb{R})$. For multiresolution analysis, the Haar scaling function is given by (see figure 2.4)

$$\phi_{\text{Haar}}(t) = \phi_{[0,0]}(t) := \begin{cases} 
1, & 0 \leq t < 1 \\
0, & \text{otherwise}
\end{cases}$$

$$= 1_{[0,\frac{1}{2})} + 1_{[\frac{1}{2}, 1]}$$

Similarly we define

$$\phi_{[m,n]} := \frac{1}{2^m} \phi_{\text{Haar}} \left( \frac{t - 2^m n}{2^m} \right)$$
For discrete signals $f[n]; n \in \mathbb{Z}$, we convert the signal into a continuous form by forming a simple step function

$$f(t) = f[n] ; n \leq t < n + 1$$

$$= f[n] \phi_{[0,n]}$$

In practice, the multiresolution analysis of discrete signals are achieved by using filter banks. From theorem 2.4.7 we have

$$\phi(t) = \sqrt{2} \sum_{n=-\infty}^{+\infty} h_n \phi(2t - n)$$

$$\Rightarrow \frac{1}{\sqrt{2}} \phi \left( \frac{1}{2} \right) = \sum_{n=-\infty}^{+\infty} h_n \phi(t - n)$$
The discrete sequence \( \{h_n\}_{n=-\infty}^{\infty} \) is interpreted as a discrete filter. Discrete filters \( \{h_n\}_{n=-\infty}^{\infty} \) whose transfer functions satisfy
\[
|H(\eta)|^2 + |H(\eta + \pi)|^2 = 1
\]
are called \textit{conjugate mirror} filters, where \( H = \widehat{h} \). These conjugate mirror filters enable us to decompose a discrete signal into its different frequency bands using filter banks. From theorem 2.4.13 we have
\[
\frac{1}{\sqrt{2}} \psi \left( \frac{1}{2} \right) = \sum_{n=-\infty}^{\infty} g_n \phi(t - n)
\]
\( \{g_n\}_{n=-\infty}^{\infty} \) is also a discrete filter. The coefficients of a signal, decomposed using an orthonormal wavelet basis, are computed using a fast algorithm that uses discrete convolutions of subsamples of the signal with the filters \( \{h_n\}_{n=-\infty}^{\infty} \) and \( \{g_n\}_{n=-\infty}^{\infty} \).

Figure 2.6: The Haar Wavelet Transform applied to the signal in figure 2.5

Given a function \( f \in L^2(\mathbb{R}) \), the orthogonal projection of \( f \) on space \( V_i \) is defined as
\[
P_{V_i} f = \sum_{n=-\infty}^{\infty} \langle f, \phi_{[i,n]} \rangle \phi_{[i,n]}
\]
Similarly, the orthogonal projection of $f$ on space $W_i$ is given by

$$P_{W_i}f = \sum_{n=-\infty}^{\infty} \langle f, \psi_{i,n} \rangle \psi_{i,n}$$

Since the space $W_i$ is the orthogonal complement of the space $V_i$, we have

$$P_{V_i}f = P_{V_{i+1}}f + P_{W_{i+1}}f$$

The fast wavelet transform (FWT) accomplishes just that. $P_{W_{i+1}}f$ gives us the *details* (local fluctuations) of $f$ that appear at the scale $2^i$ but disappear at coarser levels, e.g. $2^{i+1}$. Reconstruction from the decomposition coefficients gives us $P_{V_{i+1}}f$ from the coarser approximation given by $P_{V_{i+1}}f$ and $P_{W_{i+1}}f$. Let us define the notation $x[n] := x_n$, $\bar{x}[n] := x[-n]$. We also define a notation for even-subsampling as

$$\bar{x}[n] := \begin{cases} x[k], & \text{if } n = 2k \\ 0, & \text{if } n = 2k + 1 \end{cases}$$

where $n, k \in \mathbb{N}$. Let

$$a_i[n] := \langle f, \Phi_{i,n} \rangle$$
$$d_i[n] := \langle f, \Psi_{i,n} \rangle$$

**Theorem 2.4.14** (refer [Mal99])

*The coefficients of the decomposition are given by*

\[
a_i[k] = \sum_{n=-\infty}^{\infty} h[n-2k] a_{i-1}[n] \\
= a_{i-1} \ast \bar{h}[2k] \tag{2.4.9}
\]

\[
d_i[k] = \sum_{n=-\infty}^{\infty} g[n-2k] d_{i-1}[n] \\
= d_{i-1} \ast \bar{h}[2k] \tag{2.4.10}
\]

*(see figure 2.7)*

![Wavelet Decomposition](image)

Figure 2.7: Wavelet Decomposition
The reconstruction step is given by

\[ a_{i-1}[k] = \sum_{-\infty}^{\infty} h[k - 2n] a_i[n] + \sum_{-\infty}^{\infty} g[k - 2n] d_i[n] \]

\[ = \tilde{a}_i * h[k] + \tilde{d}_i * g[k] \] (2.4.11)

(see figure 2.8)

Filter \( \tilde{h} \) acts as a low-pass filter and \( \tilde{g} \) as a high-pass filter.

Figure 2.5 gives an example of a one dimensional signal. Figure 2.6 displays a single step of the wavelet decomposition of the same signal using the discrete Haar scaling and wavelet functions. For a two dimensional signal, like an image (figure 2.9 is an example), the wavelet transform is applied cumulatively one dimension at a time. For example, the two-dimensional wavelet transform may be computed by applied the one-dimensional transform on each of the rows of the image. Each row or column of image pixel values can be viewed as being a signal by itself (see figure 2.10). The one-dimensional transform is then applied columnwise upon the results attained from the previous step. This process completes a two-dimensional wavelet transform for an image (see figure 2.11). This process can also be defined in terms of tensor products (see [Nie99, Mal99]).

The next chapter describes an algorithm that used the two-dimensional Haar transform to obtain a representative measure or signature with which to describe the texture content of a given image (or a section thereof).
Figure 2.9: A sample image

Figure 2.10: The Haar Wavelet Transform applied on the rows of the previous image.
Figure 2.11: The Haar Wavelet Transform on the columns of the row-transformed image.
Chapter 3

Multiresolution Content Based Image Query

The basic algorithm for matching a query image $Q$ and a target image $T$ with $N$ components, as detailed in [CEJS95] is as follows:

1. Process the wavelet-based measure from the query image $Q$ (see figure 3.1).
   
   (a) Perform a complete wavelet decomposition using the Haar wavelet on the query image yielding the matrix $W_Q$
   
   (b) Store the average value from the query decomposition. Let $V_{WQ}(1) = W_Q(1,1)$.
   
   (c) From the rest of the matrix $W_Q$ store the signs of the $N-1$ highest magnitude coefficients into the vector $V_{WQ}(k), k \in \{2, \cdots, N\}$ in a descending order based upon their
magnitudes. Their positions (row and column information) within the matrix are stored in the position matrix \( P_{WQ} \). \( P_{WQ}(k,1) \) stores the row value for the \( k \)th highest magnitude detail coefficient. Similarly \( P_{WQ}(k,2) \) holds the column value. The vector-pair \((V_{WQ}, P_{WQ})\) is the wavelet-based measure for the query image.

2. Process the wavelet-based measure from each target image \( T \) from the archive (see figure 3.1).
   
   (a) Perform a complete wavelet decomposition on the target image yielding the matrix \( W_T \).
   (b) Store the average value from the target decomposition. Let \( V_{WT}(1) = W_T(1,1) \).
   (c) From the rest of the matrix \( W_T \) store the signs of the N-1 highest magnitude coefficients into the vector \( V_{WT}(k), k \in \{2, \cdots, N\} \) in a descending order based upon their magnitudes. Their positions (row and column information) within the matrix are stored in the position matrix \( P_{WT} \). \( P_{WT}(k,1) \) stores the row value for the \( k \)th highest magnitude detail coefficient. Similarly \( P_{WT}(k,2) \) holds the column value. The vector-pair \((V_{WT}, P_{WT})\) is the wavelet-based measure for the target image.

3. Compute \( W(Q, T) \), a metric that computes the difference between the wavelet-based measures of the query and the target images. The metric is given by

\[
W(Q, T) = w_1 |V_{WQ}(1) - V_{WT}(1)| - \sum_{k=2}^{N} w_k \delta_{P_{WQ}(k),P_{WT}(k)} \delta_{V_{WQ}(k),V_{WT}(k)}
\]

and will be explained in the next section. A modification to this metric will also be discussed.

4. The value \( W(Q, T) \) is computed for all the target images in the archive. A smaller value of \( W(Q, T) \) signifies a better match.

Step 2 is usually a pre-processing step and is computed in advance. Note that this algorithm stores and uses positional information, hence matches are restricted to query and target images of the same size, and with features in the same position and orientation. Some variation is allowed since we are looking at a reduced set of components as compared to a pixel-by-pixel approach.

### 3.1 The Wavelet Metric

The metric used to compare a query image \( Q \) and a target image \( T \), as discussed in [CEJS95], is given by:

\[
W(Q, T) = w_1 |V_{WQ}(1) - V_{WT}(1)| - \sum_{k=2}^{N} w_k \delta_{P_{WQ}(k),P_{WT}(k)} \delta_{V_{WQ}(k),V_{WT}(k)}
\]

(3.1.1)

where

\[
\delta_{P_{WQ}(k),P_{WT}(k)} = \begin{cases} 
1 & \text{if } P_{WQ}(k,1) = P_{WT}(k,1) \text{ and } P_{WQ}(k,2) = P_{WT}(k,2) \\
0 & \text{otherwise}
\end{cases}
\]
and
\[ \delta_{VwQ(k),VwT(k)} = \begin{cases} 
1 & \text{if } VwQ(k) = VwT(k) \\
0 & \text{otherwise}
\end{cases} \]

The \( w_k \) are weights that are pre-determined during a training phase.

### 3.2 Training the Weights

The previous section describes the process of obtaining the wavelet metric for a Query-Target pair \((Q,T)\). The expression for the metric includes a set of variables namely \( w_1, w_2, \ldots, w_n \), also known as the weights for the metric. It is these values that give the metric its meaning. These decide how well a \((Q,T)\) pair matches each other. These values are not known beforehand. A good way to estimate them is to determine them from a preselected set of \((Q,T)\) pairs that is fairly representative of the type of images that one is looking to work with. By type we refer to the overall composition of the image. For example one type of image may contain a small number of relatively homogeneous objects. Another type of image may contain a high level of detail. Yet another type may consist of those images that consist of just an outline of the boundaries of the objects in a scene.

It is good to note here that we do not expect one set of weights to work optimally for all kinds of images. For example, images with locally homogeneous components, images with outlines and images with a high degree of detail would typically be represented by different sets of weights. Ideally of course we would like one set of weights to be able to perform matches on all image types. But quite intuitively, different sets of weights focus on specific image categories and hence give focussed results. The set of preselected \((Q,T)\) pairs for a specific image type is called the training set.

Suppose that the training set is has \( m \) Query images and \( n \) Target images. For ease of training the \((Q,T)\) pairs are chosen such that a query matches only one target. Suppose, for example, that Query \( Q_i \) matches Target \( T_j \). The decision concerning which pairs form a match and which do not, is a manual process. Let \( 1 \) denote a match and \( 0 \) denote a mismatch. Then for each Query \( Q_i \), we have equations of the form:
\[
M(Q_i, T_1) = 0 \quad M(Q_i, T_2) = 0 \quad \cdots \quad M(Q_i, T_j) = 1 \quad \cdots \quad M(Q_i, T_n) = 0,
\]

where \( M(Q_i, T_j) \) refers to the expression for the metric as detailed in the previous section. Thus, in all, we have \( m \times n \) such equations in \( N \) unknowns (the weights). \( N \) is an arbitrary number that is empirically selected based upon the performance of the algorithm for images of that particular type. Values of \( N \) varying from 40 to 80 have been tested and give varying results. The original paper [CEJS95] discusses the effect of this variable on the results. It also suggests a binning method that reduces the number of unknown weights in the equations. The values of the weights can be obtained using the least squares approximation method. The number of training images required to obtain an effective set of weights is empirically determined.
3.3 The Logit Model for Least Squares

This method for obtaining the weights was suggested in appendix of the original work [CEJS95]. This section describes the method in brief. Notice, that we want the metric to give us a binary result: 1 for a match and 0 for a mismatch. In the previous section we suggest the same for training the weights. But the weights thus obtained by a least squares approach do not guarantee a binary result when applied to a test (Q,T) pair. In fact the least squares method does not even guarantee that the result would lie within the range [0,1]. The logit model essentially maps the range \((0,1) \rightarrow (-\infty, \infty)\) on a logarithmic scale. Let \(L(x) = \log(\frac{x}{1-x})\) where \(x \in (0,1)\) (see figure 3.3). The graph decreases very rapidly to \(-\infty\) as \(x \rightarrow 0\) and increases as rapidly to \(\infty\) as \(x \rightarrow 1\).

Thus to imitate a binary representation for matches and mismatches, we may for example, choose a large positive value (say +200) for \(L(x)\) to represent a match and \(L(x) = -200\) for a mismatch (see figure 3.4). These values may then be used in place of the 1’s and 0’s in the equations for the metric calculations on the training set. The weights thus obtained from the least squares approximation will induce the result of any metric calculation to always lie in the range \((-\infty, \infty)\). But, now this result can be easily mapped back to the \((0,1)\) range. Let \(y = L(x)\). Then \(x = \frac{e^y}{1+e^y}\). One can binarize the result using a threshold value. For example, \(L(x) = 0\) or \(x = 0.5\) can act as a threshold. So \(x \geq 0.5\) would imply a match and \(x < 0.5\) would denote a mismatch.

3.4 Quantization

Let us view the expression for the wavelet metric again:

\[
W(Q, T) = w_1 |V_{WQ}(1) - V_{WT}(1)| - \sum_{k=2}^{N} w_k \delta_{P_{WQ}(k), P_{WT}(k)} \delta_{V_{WQ}(k), V_{WT}(k)}
\]

where

\[
\delta_{V_{WQ}(k), V_{WT}(k)} = \begin{cases} 
1 & \text{if } V_{WQ}(k) = V_{WT}(k) \\
0 & \text{otherwise}
\end{cases}
\]
\[ L(x) = \log \frac{x}{1-x} \text{ for } x \in [0, 1] \]

Figure 3.3: \( L(x) = \log \frac{x}{1-x} \) for \( x \in [0, 1] \)

\( \delta_{vWQ(k),vWT(k)} \) here works on the quantized sign values of the magnitudes of the wavelet coefficients as outlined in steps 1(c) and 2(c) at the beginning of this chapter. Thus \( V_{WQ}(k) \in \{-1, 0, 1\} \) and so is \( V_{WT}(k) \). But such a quantization may not be limited to \( \{-1, 0, 1\} \). The original algorithm [CEJS95] used the quantization mentioned above. The results in this chapter are also based upon this very type of quantization. Various other quantizations were attempted. The results obtained were not always consistent and hence an adequate explanation was not possible.

### 3.5 Textures

The focus of this thesis is mainly on images that contain texture patterns. These are images that have a relatively high level of detail. The approaches in this chapter have been tested on the VISTEX database of texture images available at the web site: http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html. (see figure 3.5 for some examples of texture patterns available in the database.)

The original algorithm [CEJS95] was also tested on an archive of arbitrarily selected images downloaded from the internet. The components within these images were relatively homogeneous. The original algorithm worked well for these images but the results on texture images were not as good. This led to the modification to the original algorithm that will be discussed shortly. The original algorithm had another drawback. Since metric is position dependent, minor changes in the query image could throw the matches off completely. The modified version of the algorithm incorporates variation with certain positional bounds and thus provides better matches in such cases.
Notice also, that the metric is governed by the position dependent term

$$\delta_{P_{WQ}(k), P_{WT}(k)} = \begin{cases} 1 & \text{if } P_{WQ}(k, 1) = P_{WT}(k, 1) \text{ and } P_{WQ}(k, 2) = P_{WT}(k, 2) \\ 0 & \text{otherwise} \end{cases}$$

The shortcomings of the original algorithm stems from the fact that the metric uses \textit{strict} positional arguments (see figure 3.2). In other words, for a better match, there has to be a strong overlap between the query and target images starting at the top left corner, assuming of course that the average values at each level of the wavelet transform, during the computation of the metric, is stored in the upper left quadrant. Thus a small shift in the positions of a common component, in both the query and the target, can throw the metric off completely since the wavelet signature stores the quantized values of the most significant variations (due to the nature of the Haar Wavelet Transform) and their respective positions.

The modified version of this algorithm as detailed later, uses a windowing technique that takes care of small shifts in the salient variations. We are not concerned so much with large shifts, since the focus of this modified algorithm is texture images which typically will have high detail content at the local level. The original algorithm works well to match large objects whose edges are in the same position in both the query and the target. A combination of the original and the modified algorithms could therefore be used to match objects and their constituent textures. Attempts have been made in that direction but the details of that study is beyond the scope of this thesis. The wavelet transform at each level of the 2D decomposition works on the average values of the previous level.
Figure 3.5: Some sample textures from the VISTEX database.
3.5.1 Forcing Detail Coefficients

In the original algorithm, the \((N - 1)\) detail coefficients of the largest magnitude are chosen. The magnitudes are read one level at a time. So if a higher and lower level both have an identical large magnitude, we are in a quandry as to which one to pick. In the original algorithm, the choice would always be the position value at the higher level. This is especially true for texture images due to the high amount of variation detail. To provide better uniformity and incorporation of the significant details at the lower levels, we may restrict our observation to only specific levels (see figure 3.6). The choice of those levels is an empirical exercise. For the images in the VISTEK database, the first level along with the last three gave the best results.

![Figure 3.6: Limiting the algorithm to specific levels](image)

3.5.2 Windowing

Even though the query and target may be related, direct positional matches (see figure 3.7) of their wavelet signatures may be difficult to come by. To allow for some positional fluctuations, we employ a windowing technique.

![Figure 3.7: Positional Matches](image)

Every position \(P\) in the target, which does not have a direct match within the query, is matched with the position values of the transformed query in a window around the available positions \(Q\) in the wavelet signature of the query (see figure 3.8). The window size may be held fixed but, intuitively, lower levels will have a larger window size.
3.6 Results and Implementation Details

Admittedly, the search procedure described above is restrictive. It works best with images of the same size. But the modifications to the algorithm provide good results even when the query is a subset of the target. This is mainly because the images being considered here, have a lot of textural content and because the windowing and forcing of the details at specific levels compensate for minor variations and concentrate instead on the low level details. This induces the image signature to incorporate a more representative nature with respect to the images and thus improves the matching criteria.

The algorithm was tested on texture images from the VISTEX database (see figure 3.5 for examples). Both the windowing technique and the forcing of details technique were employed. The target database was a set of 110 texture images, each of size 128x128 pixels. The training set was a set of 20 images from among the targets. The number of training images required for best results is empirically determined. Of course, one would like to keep that number as low as possible to save on computation. The query images used were 64x64 pixel squares that are taken from the target images. The query images are then scaled to the same size as target images so as to have uniform positional arguments. The implementation of the algorithm ranks the target database with the query with the wavelet metric described before. It then displays the images with the lowest 9 metric-values for the match in row major order. The following are some of the results from the implementation. In all but 2 cases out of the 110, the algorithm ranked the correct target image with the lowest value, but in all 110 cases the correct target image was present among the 9 lowest ranked target images. Figures 3.10, 3.12, 3.15, 3.18, 3.21, 3.24, 3.27, 3.30 represent the 64x64 query images (smaller sections taken from the target archive). Figures ??, 3.13, 3.16, 3.19, 3.22, 3.25, 3.28, 3.31 (the image size shown is a reduced version, for display purposes) represent the 9 lowest ranked target images from the archive, corresponding to the respective queries, in row major ascending order of metric rank. Figures 3.11, 3.14, 3.17, 3.20, 3.23, 3.26, 3.29, 3.32 show the lowest ranked target image corresponding to the respective query images. Figures 3.30, 3.31 and
3.32 display the rare case where the algorithm does not rank the intended target as the lowest rank. But note that the intended target does appear in the set of 9 lowest ranked targets corresponding to that query. Note also that in all these cases (and all others), the algorithm not only picks out the intended target, but also other targets that appear similar (to the human eye). In this context it is especially interesting to note the results displayed by the figures 3.12, 3.13 and 3.14. The first four images are clearly similar (the database did not contain any others that were similar). Queries 6 and 7 are also interesting in that Target 6 is actually a subset of Target 7. The results get better with a larger training set. Overall the algorithm gives us quite satisfactory results.
Figure 3.9: Query Image 1

Figure 3.10: Ranking of Images from the Target Database, w.r.t. Query 1

Figure 3.11: Target Image with the lowest metric-rank, w.r.t. Query 1
Figure 3.12: Query Image 2

Figure 3.13: Ranking of Images from the Target Database, w.r.t. Query 2

Figure 3.14: Target Image with the lowest metric-rank, w.r.t. Query 1
Figure 3.15: Query Image 3

Figure 3.16: Ranking of Images from the Target Database, w.r.t. Query 3

Figure 3.17: Target Image with the lowest metric-rank, w.r.t. Query 3
Figure 3.18: Query Image 4

Figure 3.19: Ranking of Images from the Target Database, w.r.t. Query 4

Figure 3.20: Target Image with the lowest metric-rank, w.r.t. Query 4
Figure 3.21: Query Image 5

Figure 3.22: Ranking of Images from the Target Database, w.r.t. Query 5

Figure 3.23: Target Image with the lowest metric-rank, w.r.t. Query 5
Figure 3.24: Query Image 6

Figure 3.25: Ranking of Images from the Target Database, w.r.t. Query 6

Figure 3.26: Target Image with the lowest metric-rank, w.r.t. Query 6
Figure 3.27: Query Image 7

Figure 3.28: Ranking of Images from the Target Database, w.r.t. Query 7

Figure 3.29: Target Image with the lowest metric-rank, w.r.t. Query 7
Figure 3.30: Query Image 8

Figure 3.31: Ranking of Images from the Target Database, w.r.t. Query 8

Figure 3.32: Target Image with the lowest metric-rank, w.r.t. Query 8
Chapter 4

Markov Random Fields in Image Processing

4.1 Introduction

There are various approaches in image processing literature that focus on the extraction a measure that is representative of an image or a section of the image. Popular approaches among these are 2nd order statistics such as gray-level co-occurrence matrices, etc. It is quite an intuitive and appealing idea, when describing a texture, that the value at a pixel is influenced by those in its immediate vicinity. This approach allows us to segment the image based upon texture measures obtained from sections of the image. It also allows us to compensate for point spread effects of the imaging equipment used in the acquisition of the image. Markov Random Fields may be used to model the image for the purpose of extracting texture measures. There has been substantial work that outlines the application of Markov Random Fields to image smoothing and segmentation (see [CJ83, Dai89, DSW98, DE87, GG84, MC91, PH95, SWR/B799, SRSD98]).

4.2 Random Fields

We are familiar with the concept of a random variable, say $X$ and an associated probability distribution $P(X)$ (The choice of notation in using $P(.)$ is to emphasize the fact that, since we are dealing with images that traditionally have a discrete set of values, the probability distribution is also discrete). In the multivariate case we have $n$ random variables $X_1, X_2, ..., X_n$ whose behaviour is described by a joint probability mass function $P(X_1, X_2, ..., X_n)$. A Random Field is infact just such a set of random variables that are arranged on a lattice (See Figure 4.1).

**Notation 4.2.1** (see [Bre]) $S$ : the set of all sites of the random field.
$s$ : any single site in the random field.
$X$ : multivariate random variable representing the random field.
$X_s$ : random variable at site $s$ within the lattice.
$x_s$ : value of $X_s$
$\Lambda$ : set of possible values for $X_s$. For a 256-color image, $\Lambda = \{0, 1, 2, ..., 255\}$

Thus when a Random Field is used to model an image, the sites $S$ of the random field $X$ correspond to the pixel positions of the image. Note, however, that the image is not a random field. It is an
instance of the random field which is being used to model it. Also, the entire image need not be modelled by a single random field. Sections of the image may be modelled by separate random fields. This idea is used to extract possibly different texture measures from different regions of the image. These measures are then compared to form a segmentation. In general one thinks of a Random Field as a $m \times n$ rectangular lattice. However, the random field can have any shape. It is usual practice to consider random fields that are connected. Any variation may also be implemented, although for disconnected regions, one would usually use separate random fields. In a Random Field model, the main idea is to describe the value at a certain site in terms of the other sites. One may use a straightforward linear model to describe this interaction, or use the joint probability mass function associated with the random field.
4.2.1 Models and Energy Functions

Suppose now that we have a random field $X$. If we look at the random field only as a whole, there is only so much that we can say about its properties. If we do not know or do not assume any local interactions or independence properties of the random variables at the local sites, we can at most compute the probability of occurrence (observation) of a particular instance of the random field given a number of observations. The existence of relationships between variables at local sites would help to form a model or parametrization that would serve to describe the local properties of the field. Intuitively, one thinks about interactions such as those between locally connected or at least neighboring sites. This is generally the case, although interactions between sites that are separated by a sizeable distance can also be considered.

All of these interactions are described in a model. Thus a model is a description of the relation between any site and the rest of the sites in the field. For example, a single texture would be described by a single model. There can be more than one texture present in an image. We assume here that, in such a case, that a separate model is applied to each such texture. Models may be known (as in the case where we know the details of the data acquisition mechanism, e.g. the parameters of the point spread function), or a most probable model may be selected depending on the data.

A model is described by a set of parameters $\theta$ (also referred to as weights), and a function $f(X, \theta)$ (also known as the energy function for the model). These models describe the desired/expected behaviour at each local site $s$. The global behaviour of the random field can now, in turn, be described using these local functions. Various models and their parameter estimation techniques have been presented by [Bes74, Bes86, LD89, DE87, GG84, DSW98, MC91].

4.2.2 The Linear Model

In this model, the value of a pixel at a certain site $s$ is modelled as a weighted sum of the values at the other sites.

$$X_s = f(X_{S \setminus s})$$

(4.2.1)

where

$$f(X_{S \setminus s}) = \sum_{k \neq s, k=1}^{m \times n} w_k x_k$$

(4.2.2)

and where $S$ is assumed to be a $m \times n$ array of sites, and $s$ denotes a particular site index.

This gives an intuitive feel of homogeneity. One could then, possibly, think of a least squares or least norm approximation to determine all of the weights. Thus we would be looking to minimize the $\| \cdot \|_2$ norm of the residue $r$ in the following:

$$Aw = x + r$$

(4.2.3)

where $A \in \mathbb{Z}_\Lambda^{(m \times n - 1) \times (m \times n - 1)}$, i.e. the elements of matrix $A$ take values in the integers bounded by set $\Lambda$ defined previously. Also $w \in \mathbb{R}^{(m \times n - 1)}, x \in \mathbb{Z}_\Lambda^{(m \times n - 1)}$ and $r \in \mathbb{R}^{(m \times n - 1)}$. 

50
The hurdle that is clearly evident here is the fact that even for a relatively small sized image, say 100 × 100, A is a 9999 × 9999 sized matrix. Clearly the least squares computation goes out of hand fairly quickly.

Now, intuitively, we may think of restricting the dependence of a pixel to those pixels that are in closer proximity. This is in fact a fact a viable option that reduces the matrix size for the least squares computation and is sometimes used as a first step in the computation of the weights for image segmentation (see [MC91]). However, the resultant weights reflect the properties of the local neighborhood that was chosen for the computation, and not the image and a whole. A bottom up approach may be applied to stitch the weights from the different regions into that for the whole image. The advantage of this approach however is its straightforward deterministic computation.

The choice of a model rests largely with the application for which it is being utilized. Nonlinear models may, of course, be utilized. For example, let \( (i, j) \) be the index into the \((i, j)\)th location in the lattice \( S \), and let the index of site \( s \) be \((a, b)\). An example of a non-linear model function is:

\[
 f(X_s) = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{(i,j)} x_{k_{\text{max}}\left|i-j\right|}; (i, j) \neq (a, b)
\]

The model above implies that the effect of the other pixels on \( s \) decreases exponentially with the distance from \( s \).

### 4.2.3 The Degradation Model

Simply put, let \( x \) be an instance of \( X \).

\[
 \text{Let } \mu = \frac{1}{|S\setminus s|} \sum_{k \in S\setminus s} x_k
\]

where \( |S\setminus s| \) denotes the number of sites in the lattice \( S \) minus the site \( s \) being considered. Then:

\[
 X_s - \mu = \sum_{k \neq s, k \in S} w_k (x_k - \mu)
\]

This model works on differences from the mean and thus models variations better.

### 4.2.4 Stochastic Models

The effectiveness of the linear model (degradation or normal) is limited by the size of the image. In the event that the entire image is to be modelled as a single random field, the large size of the coefficient matrix makes the least squares (or least norm) computation expensive. We are looking for a set of weights (henceforth also called measures or parameters) that model the entire image or even a sizeable region within the image. With that in mind and the fact that if we can establish some kind of independence between regions then a probability distribution can be used as a modelling function so that the overall joint probability reduces to a simple multiplication. Thus the task
of estimating the best set of parameters for the model boils down to the task of finding the set of parameters that maximize the probability of those parameters given the texture data values at all the sites.

Suppose that we are given a model (i.e. parameters and an energy function) that describes the behaviour of the random field at each site \( s \). For a stochastic model, this energy function is in fact a probability density function (pdf). Thus for a general random field \( X \) let \( \vec{\theta} \) be the model parameters and let the probability density function be given by:

\[
p(X_s = x_s | X_{S \setminus s} = x_{S \setminus s}, \vec{\theta} = \vec{w}) = f(X, \vec{\theta})
\]

where \( f \) is the energy function and where \( \vec{w} = (a, b_1, b_2, b_3, ..., b_{n-1}) \) are the weights (parameter values) for the \( n \)-dimensional vector \( \vec{\theta} \) of parameters of the model. For ease of representation, the above pdf will also be represented as \( p(X_s | X_{S \setminus s}, \vec{\theta}) \).

A joint probability density function (hereafter abbreviated as pdf) for the entire random field could be determined if the relationships between the random variables \( X_s \) at each of the individual sites \( s \), with respect to each of the other random variables \( X_{S \setminus s} \), were known. But then this is rarely known and even if it were, it introduces complexity into the expression for the joint probability. The motivation for obtaining a joint probability density function for a random field is that, given an observation \( X_s \) of the random field, and a model, one can then obtain a pdf depending upon the model parameters (conditional on the particular observations). Thus we would obtain an expression of the form \( p(\vec{\theta} | X) \). We wish to find a set of parameters \( \vec{\theta} \) that will maximize the probability of \( p(\vec{\theta} | X) \). Maximizing the pdf at each individual site does not relate to the global picture. In practice, though, a simplifying assumption regarding the independence of the random variables in the random field leads to a useful representation called the pseudo-joint probability density function.

### 4.2.5 Pseudo-Joint Probability

Given the present form of the joint pdf, the energy function \( f(X, \vec{\theta}) \) clearly depends upon too many variables. For ease of expression and computation, some simplifying assumptions are introduced. These assumptions are not arbitrary (as will be subsequently explained) and they also have an intuitive appeal. The first of these is the assumption of independence. The pdf at a site \( s \) in a random field is given by \( p(X_s | X_{S \setminus s}) \). The independence assumption states that \( X_s \perp X_t \), \( \forall s \neq t \) such that \( s, t \in S \). Thus the joint pdf is now given by:

\[
p(X | \vec{\theta}) = \prod_{s \in S} p(X_s | X_{S \setminus s}, \vec{\theta})
\]

This is known as the pseudo-joint probability density function (see [MC91, Bre]). Maximum likelihood (ML) approaches for parameter estimation when applied to such an expression, are known as Maximum Pseudo-Likelihood (MPL) methods.
4.2.6 MAP

The primary objective of random field modelling is the estimation of the parameters of the model that best describes the instance of the random field that we wish to model. In other words, we wish to obtain an estimate $\hat{\theta}$ for $\theta$ that maximizes the probability of $\theta$ having the value $\hat{\theta}$, given an observation $x$ of the random field $X$. Thus

$$\hat{\theta} = \arg\max_{\theta} P(\theta | X)$$  \hspace{1cm} (4.2.6)

But then $P(\theta | X)$ is not observed directly. We only observe instances $x$ of the random field $X$. So for a particular set of parameters $\theta$, and an observation $x$ of the random field $X$, $P(X_s = x_s \mid X_{S\setminus s} = x_{S\setminus s}, \theta)$ is a prior (also called a priori) probability distribution specified by the model. Thus the pseudo-joint probability

$$P(X \mid \theta) = \prod_{s \in S} P(X_s = x_s \mid X_{S\setminus s} = x_{S\setminus s}, \theta)$$

is also observable or available a priori. $P(\theta)$ is usually assumed to have a uniform distribution. It is a reasonable assumption to make. Each configuration of the values of $\theta$ is thus equally probable. Upon applying Bayes Rule we obtain:

$$P(\theta \mid X) = \frac{P(X \mid \theta) P(\theta)}{P(X)}$$

$$P(X \mid \theta) = \frac{P(X \mid \theta) P(\theta)}{P(\theta)}$$

$$\Rightarrow P(\theta \mid X) = \frac{P(X \mid \theta) P(\theta)}{P(X)}$$  \hspace{1cm} (4.2.7)

$P(\theta \mid X)$ is called the posterior (or a posteriori) probability distribution. Recall, that we wish to find

$$\hat{\theta} = \arg\max_{\theta} P(\theta \mid X)$$

which is the value of $\theta$ that maximizes the a posteriori distribution. Thus

$$\hat{\theta} = \arg\max_{\theta} \left[ \frac{P(X \mid \theta) P(\theta)}{P(X)} \right]$$  \hspace{1cm} (4.2.8)

$$= \arg\max_{\theta} P(X \mid \theta)$$  \hspace{1cm} (4.2.9)

, since $P(X)$ does not depend upon $\theta$. Also, as stated previously $P(X \mid \theta)$ is usually modelled as a prior distribution and $P(\theta)$ is usually assumed to be uniformly distributed. Thus $\hat{\theta}$ is the MAP (Maximum A Posteriori) estimate of $\theta$. 

53
4.3 Neighborhoods

Upto this point we have considered the joint probability as a function of all the sites in the lattice. Even with the assumption of independence, the pseudo-joint probability is expressed as a product $\prod_{s \in S} P(X_s | X_{S \setminus s}, \theta)$ of conditional probabilities, where $P(X_s | X_{S \setminus s}, \theta)$ at a local site $s$ is conditional on all the other sites in $S$. Even though the pseudo-joint probability is expressed as a simple product of conditional probabilities at local sites, it becomes increasingly difficult to obtain these probability values in practice, as the size of the lattice increases.

What is desired is a smaller, and preferably fixed, number of conditional variables for each site, such that the joint and pseudo-joint probabilities at the local sites still hold. We propose the concept of a Neighborhood Structure (also simply called a Neighborhood). The idea here is that given a certain Neighborhood structure around a certain site, the conditional probability distribution for that site is a function of the sites, and their parameters, contained within that neighborhood structure.

Definition 4.3.1 Neighborhood

A Neighborhood Structure (or simply a Neighborhood) $N_s$ for a site $s$ is a set of sites such that:

1. $s \notin N_s$

2. $\forall i \in N_s, s \in N_i$

For example, one could consider a 3x3 window around site $s$, and excluding site $s$, to be a Neighborhood for site $s$ (See figure 4.3). The above example of a 3x3 window is very familiar neighborhood structure in image processing. Note, however, that the neighborhood structure need not always be a $(2n+1) \times (2n+1)$ window, $n \in \mathbb{N}$. The requirement for symmetry in a Neighborhood Structure is implicit in the definition (See figure 4.4). Neighborhoods of two different sites may overlap. The
Neighborhood Structure for a Random Field is usually the same over all the sites and is represented by $N$.

Although Neighborhood Structures of the type shown in figure 4.4 are valid structures as per the definition of a Neighborhood, in practice however, configurations of the types shown in figure 4.8 are more popular (see [Bre, MC91, SRSD98]), and we will restrict our discussion to Neighborhoods of only this type.

In our discussion we have implicitly assumed that the sites of the random field are uniformly spaced. This usually the case when we model images as random fields, since images are treated as lattices of uniformly spaced pixels (sites of the random field). This discussion can be readily extended to the case where the sites are irregularly spaced (see figure 4.6 and refer to [Li01]). In all of the discussion that follows, a Neighborhood Structure $N_s$ will also have, associated with it, a variable $r$ which redefines the Neighborhood Structure as follows:

$$N_{s,r} = \{ i \in S \mid [\text{dist}(i,s)]^2 \leq r \}; r \in \mathbb{Z}$$

(4.3.1)

where the function $\text{dist}$ gives us the Euclidean distance between the two sites in question.

The above expression 4.3.1 is a general definition that also covers the cases where the sites are irregularly spaced. In image processing applications, one generally assumes the horizontal or
vertical distance between any two adjacent pixels to be of unit length. Figure 4.7 displays the increasing sequence of neighborhoods and their corresponding values for \( r \), the radius. Notice that the values for \( r \) do not vary in unit increments. For this reason, and since hereafter we will be focussing only on regularly unit spaced sites, we introduce the concept of a Neighborhood Order.

### 4.3.1 Neighborhood Order

A Neighborhood \( N_s \) around a site \( s \) is said to be of order \( n \), if it contains all the sites within a radius \( r \) (except, of course, site \( s \) itself), where \( r \) is the \( n \)th term in a monotonically increasing sequence of integers each of which is a sum of squares of two positive integers. We always consider \( r > 0 \). In other words, a Neighborhood \( N_s \) around a site \( s \) is said to be of order \( n \), if it contains all sites (excluding \( s \)) with a label less than or equal to \( n \) in figure 4.8.

### 4.3.2 Cliques

**Definition 4.3.2 Clique**

Given a random field \( X \) and a neighborhood structure \( N_s \) with order \( n \), a clique is defined as a set \( C \) of sites of \( X \) such that \( \forall \{ s_i \} \in C \) such that \( s_i, s_j \in C \) and \( i \neq j \), we have \( s_i \in N_j \). In other words, every site in a clique contains all of the other members of the clique within the neighborhood around it.

The number and type of cliques that include a certain site \( s \) depends upon the neighborhood structure that is being used. The concept of a clique is used in the definition of Gibbs Randoms Fields in the following section.
Figure 4.7: Neighborhoods and their corresponding values for $r$.

Figure 4.8: Neighborhoods Orders.
4.4 Gibbs Random Fields

Definition 4.4.1 Gibbs Random Field
A Random Field $X$ with a Neighborhood Structure $N_s$ is a Gibbs Random Field (GRF) if and only if each of the instances of the Random Field is distributed according to a Gibbs Distribution whose energy function is based upon clique potentials (see section 4.4.1) derived from the Neighborhood Structure $N_s$.

Definition 4.4.2 Gibbs Distribution
A Gibbs Distribution on a set of random variables $\{X_s\}_{s \in S}$ is defined as:

$$P(X = x) = \frac{1}{Z} e^{-\frac{1}{T}E(x)}$$

(4.4.1)

where

$$Z = \sum_{x \in \mathbb{X}} e^{-\frac{1}{T}E(x)}$$

(4.4.2)

is a normalizing constant (also called partition function) that helps to restrict the values of $P(X)$ within the range $[0, 1]$. $\mathbb{X}$ is the set of all possible instances of the set of random variables $\{X_s\}_{s \in S}$. Recall that $X_s$ takes values from the discrete set $\Lambda$. For gray-scale images $\Lambda = \{0, 1, ..., 255\}$. $T$ is a positive constant often referred to as the temperature. $T$ is usually assumed to be 1. $T$ plays a useful role in parameter estimation via Simulated Annealing approaches (see section 4.6.3). Notice that for large values of $T$ (i.e. $T \to \infty$), $P(X) \to$ the Uniform Distribution. Also, for very small values of $T$ (i.e. $T \to 0$), $P(X) \to$ the delta distribution $\delta_X$. Thus local extrema of the function $P(X)$ become more pronounced as the temperature $T$ is decreased from a higher value to a lower one. It is essentially an approach that attempts to reach a global extrema. $E(x)$ is also called the energy function.

It is interesting to note that right hand side of equation 4.4.1 represents an exponential family of functions. Thus (see [?]erin1) a wide variety of distributions can be modelled as a Gibbs Distribution. As a very simple case consider a discrete version of the Normal Distribution $N(\mu, \sigma)$ (the Gaussian function with some mean $\mu \in \Lambda$ and variance $\sigma^2$). Some clarification is in order here. The standard Normal Distribution is in fact a continuous distribution and it is difficult to obtain an analytical expression for a definite integral over a finite interval on the real line. The discrete version here refers to any approximation of the continuous version. Here we consider an approximation which equals (upto a constant) the values of the continuous version, on the discrete domain. Let the number of sites in the lattice $|S| = 1$ (that is, we are considering only a univariate distribution). Thus the probability mass function for this discrete Normal Distribution is given by

$$P(n) = \alpha \ G(n - \mu)$$

(4.4.3)

where

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$
is the standard Normal Gaussian function, and

\[ \alpha = \frac{1}{\sum_{n \in \Lambda} G(n - \mu)} \]

is a normalizing constant. (Note that the dummy parameter \( n \) in the above equations is used just to symbolize the fact that \( P \) is in fact a probability mass function). The discrete distribution also has to satisfy the properties of a discrete probability distribution, namely

1. \( P(n) \geq 0 \ \forall n \in \Lambda \), and
2. \( \sum_{n \in \Lambda} P(n) = 1 \)

This is readily observed, since clearly \( G(n - \mu) \geq 0 \). Also, note that \( \mathbb{1}(X_s) \) at a single site \( s \) is the same as the discrete set \( \Lambda \) of possible values that \( X_s \) may assume.

**Claim:** The univariate discrete Normal Distribution is a Gibbs Distribution.

**Proof:**

For \( X_s = x \), \( x \in \Lambda \), let

\[ E(x) = \frac{(x - \mu)^2}{2\sigma^2} \]  

(4.4.4)

Recall, from property 2 above, that

\[ \sum_{x \in \Lambda} P(x) = 1 \]

\[ \Rightarrow \sum_{x \in \Lambda} \alpha G(x - \mu) = 1 \]

\[ \Rightarrow \sum_{x \in \Lambda} \alpha \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} = 1 \]

\[ \Rightarrow \frac{\alpha}{\sqrt{2\pi\sigma}} \sum_{x \in \mathbb{1}(X_s)} e^{-E(x)} = 1 \]

\[ \Rightarrow \frac{\alpha}{\sqrt{2\pi\sigma}} Z = 1 \]  , (from equation 4.4.2)

\[ \Rightarrow Z = \frac{\sqrt{2\pi\sigma}}{\alpha} \]  

(4.4.5)

Hence, from equation 4.4.3, we get

\[ P(X_s = x) = \alpha G(x - \mu) \]

\[ = \frac{\alpha}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \]

\[ = \frac{1}{Z} e^{-E(x)} \]  

(4.4.6)
using equations 4.4.4 and 4.4.5. Now, comparing equation 4.4.6 to equation 4.4.1, we see that
\( P(X_s) \) gives us a Gibbs Distribution with \( T = 1 \). For a further discussion on the equivalence between the Gibbs Distribution and other distributions, see section 4.5.3.

The value of \( Z \) is central to the definition of the Gibbs Distribution. But it is not always trivial to compute the value of \( Z \), especially in the multivariate case. Each site can assume \(|\Lambda|\) different values from the set \( \Lambda \). If there are \(|S|\) sites in the lattice, \( x \) can have \(|\Lambda|^{|S|}\) different configurations. Thus for a small gray-scale image with \( 16 \times 16 \) pixels, being modelled as a random field, \( x \) can have \( 256^{256} \) different configurations. Thus obtaining the value of \( Z \) for even a small lattice involves a lot of computation. It turns out that for the purposes of parameter estimation using a MAP based (see section 4.2.6) approach, one can do without the value of \( Z \) (see section 4.6).

A Gibbs Random Field is a Random Field with a Gibbs distribution whose energy function \( E(x) \) is based upon values (known as potentials) derived from all the cliques (and only the cliques) present in the lattice. The cliques are determined from the Neighborhood Structure \( N_s \) for the Random Field.

### 4.4.1 Clique Potentials

Given an instance \( x \) of a Gibbs Random Field \( X \) with a Neighborhood Structure \( N_s \), the Energy function \( E(x) \) is a function of all the cliques present in the lattice. Each clique has, associated with it, a value called the clique potential. This value depends upon the instance \( x \) of the Random Field. Thus the value for each clique on a given instance of the Random Field, is given by a clique potential function of the type

\[
V : C \times \mathcal{X} \longrightarrow \mathbb{R}
\]

where \( C \) is the set of all cliques in the lattice of the Random Field. Recall that \( \mathcal{X} \) is the set of all possible instances of the Random Field.

**Definition 4.4.3** \( V(c, x) \) is called the clique potential at clique \( c \in C \) when \( x \in \mathcal{X} \) is the given instance of the Random Field.

The energy function for the Gibbs Distribution in a Gibbs Random Field is given by

\[
E(x) = \sum_{c \in C} V(c, x)
\]

As a simple example, consider a first order Neighborhood on a \( 3 \times 3 \) Random Field (see figure 4.9). There are 5 cliques. An example of the clique potential function \( V \) could be the following:

\[
\begin{align*}
V(c_1, x) &= \alpha a_s \\
V(c_2, x) &= -\beta_1 a_{12} \\
V(c_3, x) &= -\beta_2 a_{32} \\
V(c_4, x) &= -\beta_3 a_{21} \\
V(c_5, x) &= -\beta_4 a_{22}
\end{align*}
\]
where $\alpha, \beta_1, \beta_2, \beta_3, \beta_4 \in \mathbb{R}$ are constants. Thus the energy function becomes

$$E(x) = \alpha a_x - \beta_1 a_{12} - \beta_2 a_{32} - \beta_3 a_{21} - \beta_4 a_{22}$$

The choice of the clique potential function $V$ is a modeling decision. For most image processing applications involving Random Fields, the following 2 properties of the clique potential are generally assumed:

1. Homogeneity:
   A Gibbs Random Field is said to be *homogeneous* if the clique potential $V(c, x)$ is independent of the location of the clique $c$ within the lattice.

2. Isotropy:
   A Gibbs Random Field is said to be isotropic if the clique potential $V(c, x)$ is independent of the orientation of the particular clique.

### 4.4.2 Local Conditional Probabilities for a GRF

Recall that it is computationally prohibitive to obtain the value of the normalizing constant $Z$ in the joint probability formulation (see equation 4.4.1) of the Gibbs Distribution, for even a small lattice. Note that $X$ has $|S|$ sites and is hence usually multivariate. Obtaining the normalizing constant becomes computationally much more tractable if one considers conditional probabilities at each of the local sites $s \in S$. If in addition, we assume the independence of the random variables at each site with respect to those at all the other sites, we can then formulate an expression for the pseudo-joint probability (see section 4.2.5) as an approximation to the actual joint probability. The local conditional probability at a site $s$ in a Gibbs Random Field is given by

$$P(X_s = x_s \mid X_{S \setminus s} = x_{S \setminus s}) = \frac{1}{Z_s} e^{-\frac{1}{\beta} E_s(x)} \quad (4.4.7)$$

where $Z_s$ is again a normalizing constant

$$Z_s = \sum_{x_s \in \Lambda} e^{-\frac{1}{\beta} E_s(x)} \quad (4.4.8)$$
Note that $x = (x_s, x_{S \setminus s})$ and

$$E_s(x) = \sum_{c \in C \text{ such that } s \in c} V(c, x) \quad (4.4.9)$$

As a general rule for the local Energy function $E_s(x)$, $V(c, x) = 0$ if $s \not\in c$. Let $C(A)$ denote all the cliques present in the set of sites $A$, and let $\bar{N} = N_s \cup s$. Then the above expression for $E_s(x)$ becomes

$$E_s(x) = \sum_{c \in C(\bar{N}) \text{ such that } s \in c} V(c, x) \quad (4.4.10)$$

Thus

$$E_s(x) = E_s(x_{N_s}) \quad (4.4.11)$$

The pseudo-joint probability is given as in equation 4.2.5. Note that in the preceding discussion we have assumed that $\vec{\theta} = \emptyset$, the empty set. Extra parameters $\vec{\theta}$ can easily be added to the expressions listed above. Thus

$$P(X \mid \vec{\theta}) = \prod_{s \in S} P(X_s \mid X_{S \setminus s}, \vec{\theta})$$

$$= \frac{1}{\prod_{s \in S} Z_s} \exp \left( -\sum_{s \in S} \frac{1}{\alpha_s} \bar{E}_s(x, \vec{\theta}) \right) \quad (4.4.12)$$

This gives us an expression for the prior distribution which can then be used to compute the MAP probability (refer to section 4.2.6). The next section discusses the concept of Markov Random Fields and their equivalence to the Gibbs Random Fields via the very useful Hammersley-Clifford Theorem.

## 4.5 Markov Random Fields

**Definition 4.5.1** A Random Field $X$ with a Neighborhood Structure $N$ is called a Markov Random Field if it satisfies the following properties:

1. **Positivity**:
   $$P(X = x) > 0 \quad \forall x \in \mathbb{X}(X)$$

2. **Markovianity**:
   $$P(X_s = x_s \mid X_{S \setminus s} = x_{S \setminus s}) = P(X_s = x_s \mid X_{N_s} = x_{N_s})$$

The positivity property just reflects the idea that every instance of the Random Field is probable. The Markovianity property goes on to say that the value at a site is dependent on only a certain region of influence (the Neighborhood) as opposed to being dependent upon all the sites in the lattice.
4.5.1 Hammersley-Clifford Theorem

The previous sections describe Gibbs Random Fields and Markov Random Fields. The Gibbs Random Fields define a local probability distribution (the Gibbs Distribution) at each site in the lattice in terms of an energy function calculated over cliques present in the entire lattice. Markov Random Fields are Random Fields that have a positive probability and obey the Markovianity property. A Random Field that obeys the Markovianity property has a well defined neighborhood structure such that the probability at a local site with respect to all the other sites in the lattice actually depends only on the sites in the neighborhood of that site.

The Gibbs Random field has a neighborhood structure and specifies probabilities in terms of a function of all the cliques present in the lattice. The Markov Random Field on the other hand obeys the Markovianity property, but does not specify a distribution. Fortunately, an (unpublished) theorem called the Hammersley-Clifford theorem (named after the original authors of the theorem) ties the two concepts together. The proof of this theorem can be found in [Bes74]. A lucid proof by Grimmett [Gri73] appeared at about the same time. The next section lays out the proof as described by [Gri73] and [Bre]. The Hammersley Clifford theorem is also known as the Gibbs-Markov equivalence theorem. It establishes the equivalence between Markov Random Fields and Gibbs Random Fields.

**Theorem 4.5.1 Hammersley-Clifford**

A Random Field is a Markov Random Field $\Leftrightarrow$ the probability distribution associated with the Random Field is a Gibbs distribution.

In other words:

$$P(X_s | X_{S \setminus s}, \overrightarrow{\theta}) = P(X_s | X_{N_s}, \overrightarrow{\theta})$$

if and only if

$$P(X | \overrightarrow{\theta}) = \frac{1}{Z} e^{-E(X, \overrightarrow{\theta})}$$

where $Z$ is the partition (normalizing) function.

$$Z = \sum_{\forall x \in \Omega(X)} e^{-E(x, \overrightarrow{\theta})}$$

, where $\Omega(X)$ denotes the set of all possible instances of the random field $X$. $x$, an instance of $X$ can be represented as $(x_1, x_2, ..., x_{|S|})$ where each $x_i, i \in (1, ..., |S|)$ takes values in the set $\Lambda$ of possible values that each site may assume. Recall, from section 1.2, that for a 256-color image $\Lambda = (0, ..., 255)$ and it represents the color-range for the pixels of that image. $|S|$ denotes the number of sites present in the lattice.

In the above expression for $P(X) , E(X, \overrightarrow{\theta})$ is an energy function which is based on the cliques of the Random Field. This energy function and the parameters $\overrightarrow{\theta}$ together represent the model for the Random Field.
Proof: Hammersley-Clifford Theorem
This proof follows the approach taken by [Gri73] and later outlined by [Bre]. An alternate proof is presented in [Bes74]. Recall that the theorem states that: A Random Field is a Markov Random Field \( \iff \) the probability distribution associated with the Random Field is a Gibbs distribution. In other words we wish to prove that given a Random Field, it is a Markov Random Field iff it is a Gibbs Random Field.

Claim 4.5.1 \( \iff \) A Gibbs Random Field is a Markov Random Field.

Proof of Claim:
Let the probability distribution for the Gibbs Random Field with neighborhood structure \( N \), be given by:

\[
P(X = x \mid \vec{\theta}) = \frac{1}{Z} e^{-\frac{1}{2}E(x, \vec{\theta})}
\]

(4.5.1)
as described above. Now, using Bayes rule for conditional probability we know that

\[
P(X_s \mid X_{S\setminus s}, \vec{\theta}) = \frac{P(X_s, X_{S\setminus s} \mid \vec{\theta})}{P(X_{S\setminus s} \mid \vec{\theta})}
\]

(4.5.2)

\[
= \frac{P(X_s \mid \vec{\theta})}{P(X_{S\setminus s} \mid \vec{\theta})}
\]

(4.5.3)

By the law of Marginal Probability

\[
P(X_{S\setminus s}) = \sum_{X_s \in \Lambda} P(X_s, X_{S\setminus s}).
\]

Therefore

\[
P(X_s \mid X_{S\setminus s}, \vec{\theta}) = \frac{P(X_s, X_{S\setminus s} \mid \vec{\theta})}{\sum_{X_s \in \Lambda} P(X_s, X_{S\setminus s} \mid \vec{\theta})}
\]

(4.5.4)

In equation 4.5.1, \( E(x, \vec{\theta}) \) is the energy function over all the sites in the random field. For a Gibbs Random Field, the energy function is made up of potential functions computed over each of the cliques present in the random field. The orders of the cliques are limited by the Neighborhood structure that is being employed. Thus

\[
E(X = x, \vec{\theta}) = \sum_{c \in C(X)} V(c, x, \vec{\theta})
\]

(4.5.5)
where \( \mathbf{C}(\mathbf{X}) \) is the set of all cliques present in \( \mathbf{X} \), and \( V(c, \mathbf{x}) \) is the clique potential function as discussed in section 4.4.1. From equation 4.5.5, equation 4.5.4 and 4.5.1 we have

\[
P(X_s = x_s \mid X_{S \setminus s} = x_{S \setminus s}, \vec{\theta}) = \frac{\frac{1}{Z} e^{-\frac{1}{\beta} E(x, \vec{\theta})}}{\sum_{x_s \in \Lambda} \frac{1}{Z} e^{-\frac{1}{\beta} E(x, \vec{\theta})}} - \frac{1}{\beta} \sum_{c \in \mathbf{C}} V(c, x, \vec{\theta}) = \frac{\sum_{x_s \in \Lambda} \frac{1}{Z} e^{-\frac{1}{\beta} E(x, \vec{\theta})}}{\sum_{x_s \in \Lambda} e^{-\frac{1}{\beta} \sum_{c \in \mathbf{C}} V(c, x, \vec{\theta})}} \quad (4.5.6)
\]

We can now partition the collection of cliques \( \mathbf{C} \) into two disjoint sets, those that contain site \( s \) that is \( \mathbf{C}_s = \{ c \in \mathbf{C} \mid s \in c \} \) and those that do not, i.e. \( \mathbf{C}_s = \mathbf{C} - \mathbf{C}_s \). Therefore rewriting equation 4.5.6 we have

\[
P(X_s = x_s \mid X_{S \setminus s} = x_{S \setminus s}, \vec{\theta}) = \frac{-\frac{1}{\beta} \sum_{c \in \mathbf{C}_s} V(c, x, \vec{\theta})}{\sum_{x_s \in \Lambda} e^{-\frac{1}{\beta} \sum_{c \in \mathbf{C}_s} V(c, x, \vec{\theta})}} \quad (4.5.7)
\]

The \( e^{-\frac{1}{\beta} \sum_{c \in \mathbf{C}_s} V(c, x)} \) terms cancel out, since \( \not\exists c \in \mathbf{C}_s \) such that \( s \in c \), and hence the terms are not affected by \( x_s \). Let us now look closely at the set of all cliques based upon the Neighborhood Structure \( N \), containing the site \( s \), i.e. \( \mathbf{C}_s \). By definition (see section 4.3.2), if \( s \) belongs to a clique, then all the other sites in the clique belong to \( N_s \).

\[
s \in c \Rightarrow c \setminus \{s\} \in N_s
\]

\[
[s \in c, c \in \mathbf{C}(S)] \Rightarrow [s \in c, c \in \mathbf{C}(N_s)]
\]

where \( N_s = \{ N_s \cup \{s\} \} \). In other words, the set of cliques, in the entire lattice, that contain the site \( s \) are the same ones as the cliques that contain \( s \) in the Neighborhood \( N_s \) of \( s \). Therefore, from equation 4.5.7, we have

\[
P(X_s = x_s \mid X_{S \setminus s} = x_{S \setminus s}, \vec{\theta}) = \frac{-\frac{1}{\beta} \sum_{c \in \mathbf{C}_s} V(c, x, \vec{\theta})}{\sum_{x_s \in \Lambda} e^{-\frac{1}{\beta} \sum_{c \in \mathbf{C}_s} V(c, x, \vec{\theta})}} \quad (4.5.8)
\]

where \( \mathbf{C}_s = \{ c \in \mathbf{C}(N_s) \mid s \in c \} \). Also, note that the clique potential \( V(c, x, \vec{\theta}) = V(c, x_{N_s}, \vec{\theta}) \) \( \forall c \in \mathbf{C}_s \), since \( s \not\in \) any clique that contains any element other than those in \( N_s \).
Similarly

\[
P(X_s = x_s \mid X_{N_s} = x_{N_s}, \vec{\theta}) = \frac{P(X_{\tilde{N}_s} = x_{\tilde{N}_s} \mid \vec{\theta})}{\sum_{x_s \in \Lambda} P(X_{\tilde{N}_s} = x_{\tilde{N}_s} \mid \vec{\theta})} = \frac{\frac{1}{Z} e^{-\frac{1}{T} E(x_{\tilde{N}_s})}}{\sum_{x_s \in \Lambda} \frac{1}{Z} e^{-\frac{1}{T} E(x_{\tilde{N}_s})}} = \frac{e^{-\frac{1}{T} \sum_{c \in C(N_s)} V(c, x_{\tilde{N}_s}, \vec{\theta})}}{\sum_{x_s \in \Lambda} e^{-\frac{1}{T} \sum_{c \in C(N_s)} V(c, x_{\tilde{N}_s}, \vec{\theta})}} \tag{4.5.9}
\]

Again, we can partition the collection of cliques \(C(N_s)\) into two disjoint sets, those that contain the site \(s\) i.e. \(\tilde{\mathbf{B}}\), and those that do not, i.e. \(\bar{\mathbf{B}} = \mathbf{B} - \tilde{\mathbf{B}}\). Thus

\[
P(X_s = x_s \mid X_{N_s} = x_{N_s}, \vec{\theta}) = \frac{e^{-\frac{1}{T} \sum_{c \in \tilde{\mathbf{B}}} V(c, x, \vec{\theta})} \cdot e^{-\frac{1}{T} \sum_{c \in \bar{\mathbf{B}}} V(c, x, \vec{\theta})}}{\sum_{x_s \in \Lambda} \left( \frac{e^{-\frac{1}{T} \sum_{c \in \tilde{\mathbf{B}}} V(c, x)} \cdot e^{-\frac{1}{T} \sum_{c \in \bar{\mathbf{B}}} V(c, x)}}{e^{-\frac{1}{T} \sum_{c \in \tilde{\mathbf{B}}} V(c, x)}} \right) = \frac{-\frac{1}{T} \sum_{c \in \tilde{\mathbf{B}}} V(c, x, \vec{\theta})}{\sum_{x_s \in \Lambda} \left( \frac{-\frac{1}{T} \sum_{c \in \tilde{\mathbf{B}}} V(c, x)}{e^{-\frac{1}{T} \sum_{c \in \tilde{\mathbf{B}}} V(c, x)}} \right)} \tag{4.5.10}
\]

Comparing the above equation 4.5.10 to equation 4.5.8 we have

\[
P(X_s = x_s \mid X_{S\backslash s} = x_{S\backslash s}, \vec{\theta}) = P(X_s = x_s \mid X_{N_s} = x_{N_s}, \vec{\theta})
\]

which is the Markovianity property (see section 4.5). Also, the positivity condition for Markov Random Fields is also satisfied by every Gibbs Random Field. Hence a Gibbs Random Field is also a Markov Random Field with the same Neighborhood Structure.

**Claim 4.5.2** \((\Leftarrow): A\ \text{Markov Random Field is a Gibbs Random Field.}\)

For a proof of this claim see [Bre, Bes74].

### 4.5.2 The Exponential Family

In this section we will describe the exponential family of distributions and explain its relevance in relating various distributions to the Gibbs Distribution.

Let \(\vec{\theta} = \{\theta_i\}_{i=1}^n\) be a set of parameters. Let \(\Theta\) be the set of all possible values of \(\vec{\theta}\). Let \(\mathbf{B}(X)\) be the support for the domain of the random variable (possible multivariate) \(X\). Let \(x\) be an instance of \(X\). Now we can define functions of the type

\[
f : \mathbf{B}(X) \rightarrow (R)
\]

66
such that functions $f$ and $u_i$ never involve the variable $\theta$, and functions $g$ and $v_i$ never include any terms involving $x$.

**Definition 4.5.2 Exponential Family**

(Adapted from [CB99])

\[
P(X = x \mid \theta) = f(x) \, g(\theta) \, e^{\sum_{i=1}^{m} [u_i(x) \, v_i(\theta)]}
\]

(4.5.11)

is an expression for the $m$-parameter exponential family of distributions.

For the remainder of this discussion we will use a simplified, univariate representation for the exponential family.

\[
P(X = x \mid \theta, \phi) = e^{\left\{ \frac{\theta x + a(\theta)}{b(\phi)} - c(x, \phi) \right\}}
\]

(4.5.12)

where $a$, $b$ and $c$ are real valued functions. $\theta$ is called the cannonical or natural parameter. $\phi$ is known as the dispersion or scale parameter.

### 4.5.3 From Gibbs to Other Distributions

Recall that the Gibbs Distribution (see section 4.4) is given by

\[
P(X = x) = \frac{1}{Z} e^{-\frac{1}{T}E(x)}
\]

(4.5.13)

\[
= e^{\left( -\frac{1}{T}E(x) - \log(Z) \right)}
\]

(4.5.14)

Comparing equation 4.5.12 and equation 4.5.14 we see that we can easily express

\[
\frac{\theta x + a(\theta)}{b(\phi)} + c(x, \phi)
\]

in the form

\[
\frac{-1}{T}E(x) - \log(Z)
\]

where $T$ and $Z$ are constants as previously described in section 4.4. **Thus every instance of the exponential family is a Gibbs Distribution.** A number of commonly used distributions are part of the exponential family. Some examples of distributions that are part of the exponential family are:

- The Normal Distribution $N(\mu, \sigma^2)$
- The Binomial Distribution $B(n, p)$
- The Exponential Distribution
The Negative Binomial Distribution

The Gamma Distribution

The Poisson Distribution

An example of a distribution that is not part of the exponential family is the Uniform Distribution. We will now show explicitly that the Normal and Binomial Distributions can be viewed as Gibbs Distributions.

**Example 4.5.1** The Normal Distribution $N(\mu, \sigma^2)$ is a Gibbs Distribution

Let

$$X \sim N(\mu, \sigma^2)$$

Therefore

$$P(X = x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$= e^{-\frac{(x^2-2\mu x+\mu^2)}{2\sigma^2}-\log(\sqrt{2\pi}\sigma)}$$

$$= e^{\frac{\mu^2}{2\sigma^2}+\log(\sqrt{2\pi}\sigma)}$$  \hspace{1cm} (4.5.15)

Let $\theta = \mu$, $\phi = \sigma$. Let $a(\theta) = \theta^2$, $b(\phi) = 2\phi^2$, and $c(x, \phi) = -\left[\frac{x^2}{\sigma^2} + \log(\sqrt{2\pi}\sigma)\right]$. Then $P(X = x \mid \mu, \sigma)$ is clearly a distribution of the exponential family and thus a Gibbs Distribution.

**Example 4.5.2** The Binomial Distribution $B(n, p)$ is a Gibbs Distribution

Let

$$X \sim B(n, p)$$

Therefore

$$P(X = x \mid n, p) = \binom{n}{x} p^x (1-p)^{n-x}$$

$$= e^{\left[\log(\binom{n}{x}) + x \log p + (n-x) \log(1-p)\right]}$$

$$= e^{\left[\log(\binom{n}{x}) + x \log\left(\frac{p}{1-p}\right) + n \log(1-p)\right]}$$  \hspace{1cm} (4.5.16)

Let $\theta = \log\left(\frac{p}{1-p}\right)$ and $\phi = n$. Therefore

$$e^\theta = \frac{p}{1-p}$$

$$\Rightarrow e^\theta - pe^\theta = p$$

$$\Rightarrow e^\theta = p(1 + e^\theta)$$
Let \( a(\theta) = n \log(1 - p) \Rightarrow a(\theta) = -n \log(1 + e^\theta) \). Let \( b(\phi) = n^0 = 1 \) and \( c(x, \phi) = \log \binom{p}{x} \). Then \( P(X = x \mid n, p) \) is a distribution of the exponential family and thus a Gibbs Distribution.

### 4.5.4 Models and Energy Functions

Although one can consider a wide variety of models for a Random Field, we will restrict our discussion to the most commonly used models for image processing. As mentioned before a model is described by a set of parameters and an energy function. We represent the lattice of the Random Field by a matrix of sites (see figure 4.1).

- **The Gauss-Markov Model (GMRFs)**

Parameters (Upto a 2nd Order Neighborhood):

\[
\overrightarrow{\theta} = \{a, w_{12}, w_{13}, w_{21}, w_{23}, w_{31}, w_{33}\}
\]

Energy Function:

\[
E(x_s, x_{N_s}, \overrightarrow{\theta}) = \frac{a^2}{2} \left[ x_s - \sum_{ij} (b_{ij} w_{ij} x_{ij}) \right]
\]

When one compares this model to the Normal Gaussian function, \( \sigma = \frac{1}{a} \) and \( \mu = \sum_{ij} (b_{ij} w_{ij} x_{ij}) \). This model is also called the auto-normal model.

- **The Auto-Binomial Model**

Parameters (Upto a 2nd Order Neighborhood):

\[
\overrightarrow{\theta} = \{a, w_{12}, w_{13}, w_{21}, w_{23}, w_{31}, w_{33}\}
\]

Energy Function: We know that for a Binomial Random Variable \( X_s \sim B(n, p) \), the expectation \( \mu = \langle X_s \rangle = np \), where \( n \) for a gray-scale image is usually assumed to be the largest color index \( L \).

\[
\mu = \sum_{ij} (b_{ij} w_{ij} x_{ij})
\]

Therefore \( p = \frac{\mu}{L} \) and using equation 4.5.16 we have

\[
E(x_s, x_{N_s}, \overrightarrow{\theta}) = - \left[ \log \binom{L}{x_s} + x_s \log \left( \frac{p}{1 - p} \right) + L \log(1 - p) \right]
\]

\[
= - \left[ \log \binom{L}{x_s} + x_s \log \left( \frac{\mu}{1 - \frac{\mu}{L}} \right) + L \log(1 - \frac{\mu}{L}) \right] \tag{4.5.17}
\]
For a Gibbs Distribution, any constant term in the Energy function can be included as part of the normalizing constant $Z$. Thus

$$E(x_s, x_{N_s}, \vec{\theta}) = -\left[ \log \left( \frac{L}{x_s} \right) + x_s \log \left( \frac{\mu}{1 - \mu} \right) \right]$$

$$= -\left[ \log \left( \frac{L}{x_s} \right) + x_s \log \left( \frac{\mu}{L - \mu} \right) \right]$$

(4.5.18)

A simplified auto-binomial energy function is defined in [DSW98] and is given by:

$$E(x_s, x_{N_s}, \vec{\theta}) = -\log \left( \frac{L}{x_s} \right) - x_s \left[ a + \frac{1}{L} \sum_{ij} (b_{ij}w_{ij}x_{ij}) \right]$$

### 4.6 Parameter Estimation

Parameter estimation refers to the estimation of the parameters that best model the random field data. In other words, given an instance of a random field and a model we wish to obtain the parameters of the model that minimize the energy function. The parameters of the model that correspond to the lowest energy are said to be the characteristic features for that instance of the random field (the image) based on the assumed model. This feature vector can now be used to compare different image regions, perform image smoothing, generate textures. The Gibbs-Markov equivalence (see section 4.4) gives us

$$P(x_s \mid x_{S \setminus s} = x_{S \setminus s}, \vec{\theta}) = P(x_s = x_s \mid X_{N_s} = x_{N_s}, \vec{\theta})$$

$$= \frac{1}{Z_s} e^{-\frac{1}{T_s} E_s(x_{N_s}, \vec{\theta})}$$

(4.6.1)

where the normalizing constant $Z_s$ is given by

$$Z_s = \sum_{x_s \in \Lambda} e^{-\frac{1}{T_s} E_s(x_{N_s}, \vec{\theta})}$$

(4.6.2)

The assumption of independence of the random variables at each site with respect to all the others in the lattice, gives us the pseudo-joint probability (see Section 4.2.5). So for computational simplicity we attempt to find the MPL (Maximum Pseudo-Likelyhood) probability as opposed to the usual MLE (Maximum Likelyhood Estimate) which may be employed if in fact we are given the expression for the joint probability. Thus

$$P(X \mid \vec{\theta}) = \frac{1}{\prod_{s \in S} Z_s} e^{-\sum_{s \in S} E_s(x_{N_s}, \vec{\theta})}$$

(4.6.3)

Recall, that we wish to find the best $\vec{\theta}$ given an instance $X = x$ of the random field. Hence we are looking for a MAP estimate (See Section 4.2.6) i.e.

$$\hat{\theta} = \arg\max_{\vec{\theta}} P(\vec{\theta} \mid X)$$
If we assume that the probability distribution \( P(\theta) \) is given to be uniform, \textit{a priori}, then from equation 4.2.9 we have

\[
\hat{\theta} = \arg\max_{\theta} P(X|\theta) = \arg\max_{\theta} \left\{ \frac{1}{\prod_{s \in S} Z_s} e^{-\sum_{s \in S} \frac{1}{2} E_s(x_{\theta_s}, \theta)} \right\}
\]

\[
= \arg\max_{\theta} \left\{ e^{-\sum_{s \in S} \frac{1}{2} E_s(x_{\theta_s}, \theta)} \right\}
\]

\[
= \arg\min_{\theta} \left\{ \sum_{s \in S} \frac{1}{T_s} E_s(x_{\theta_s}, \theta) \right\} \quad (4.6.4)
\]

Various optimization techniques have been applied for the estimation of \( \hat{\theta} \) (see [CJ83, DSW98, DE87, GG84, MC91, SRSD98]). We will discuss the most commonly used approaches.

### 4.6.1 Least Squares

The minimization outlined in equation 4.6.4 can be viewed as a Least Squares fit for the set of equations

\[
\frac{1}{T_s} E_s(x_{\theta_s}, \theta) = 0 \; ; \; s \in S
\]

which when expressed in the usual form

\[
A \cdot \theta = b
\]

can be solved using the usual Least Squares formulation, i.e.

\[
\hat{\theta} = (A^*A)^T A^* b
\]

where as usual \( A^* \) is the conjugate transpose of \( A \), and \( A^T \) denotes the inverse of \( A \). The usual concerns about matrix singularity apply here. Note, however, that for the real valued \( A \), \( A^*A = A^T A \) is always symmetric. \( A^T \) is as usual the transpose of \( A \).

\[
(A^T A)^T = A^T (A^T)^T = A^T A
\]

Also, since we are modelling images using these random fields, the pixel values are almost always positive integer values. Now \( A^T A \) is atleast positive semi-definite, since

\[
x^T A^T A x = (Ax)^T Ax
\]

which is always \( \geq 0 \), since it is an inner product of two vectors.

**Claim 4.6.1** If \( A^T A \) is positive definite, then it is always invertible.
Proof of Claim:
Let $K = (A^T A)$ be a symmetric positive definite $n \times n$ matrix. Therefore
\[ x^T K x > 0, \forall x \in \mathbb{R}^n, x \neq 0 \]
Clearly $K$ cannot be a matrix of zeros, Also, $Kx \neq 0$, since then $x^T K x = 0$. Hence clearly $Kx \geq 0$ and therefore the columns of $K$ are linearly independent. $K = (A^T A)$, being a square matrix is thus non-singular and invertible.

Claim 4.6.2 A symmetric matrix $K = (A^T A)$ is positive definite if $A$ is a skinny matrix of full rank.

Proof of Claim:
A skinny matrix is one whose rows outnumber (or at most equal) its columns. If $A$ has full rank, then $Ax > 0$, $\forall x \neq 0$. Clearly then $(Ax)^T \neq 0$. Thus $(Ax)^TAx = x^TA^TAx = x^T K x > 0$. Thus $K$ is a positive definite matrix.

In image processing applications, $A$ is usually always a skinny matrix. For example when using a first order Neighborhood on a $16 \times 16$ image region, $A$ is a $256 \times 5$ matrix, since $\theta = \{a, b_{12}, b_{32}, b_{21}, b_{23}\}$ has five elements and there are 256 individual sites each of which contributes an equation. In the rare event of $A$ being a thick matrix, one may use Least Norm estimation instead (see [MC91]). This approach for parameter estimation is relatively simple to implement but the results obtained are not necessarily stable. Experimentally, however, this approach has yielded good results (see section 4.7). This approach may also be used in the initial stages of parameter estimation and the results may be further refined using more robust optimization schemes.

4.6.2 Conditional Least Squares

This concept was described in the paper by [SRSD98]. The conditional least squares (hereafter abbreviated as CLS) estimator is given by
\[ \hat{\theta} = \arg\min_{\theta} \sum_{s \in S} \left(x_s - \langle x_s \mid x_{N_s}, \theta \rangle \right)^2 \quad (4.6.5) \]
where the conditional expectation $\langle x_s \mid x_{N_s}, \theta \rangle$ is given by
\[ \langle x_s \mid x_{N_s}, \theta \rangle = \sum_{x_i \in A} x_i P(x_s \mid x_{N_s}, \theta) \]

CLS estimate for the AutoBinomial Model:
\[ \hat{\theta} = (A^T A)^{-1} Ab \quad (4.6.6) \]
where
\[ b_s = -\log \left( \frac{L}{x_s} - 1 \right) \]
\[ A_{s,j} = \begin{cases} 1, & \text{if } j = 1 \\ \frac{x_{sj}}{L}, & \text{otherwise} \end{cases} \]

where \( x_{sj} \) is the coefficient of the \( j \)th parameter in the Neighborhood of site \( s \), and \( L = \max(\Lambda) \).

**CLS estimate for the AutoNormal Model:**
This estimate can be proved to be the same as the regular Least Squares estimate.

### 4.6.3 Simulated Annealing

Other techniques for estimating the parameters of a Markov Random Field are:

- **Metropolis-Hastings**
  This method is similar to Simulated Annealing with the exception that the Energy value is also allowed to increase based upon a random probability value.

- **Iterated Conditional Modes** (see [Bes86])

- **Histogram Techniques** (see [DE87])

### 4.7 Applications to Image Processing

The previous sections discuss methods of obtaining a representative measure based on Markov Random Fields, for a given image section. The main application of the Markov Random Field based measure, for this thesis, is texture pattern matching. However the measure obtained using Markov Random Fields may be used effectively for image smoothing and texture generation applications. The results of these implementations is discussed here.

#### 4.7.1 Smoothing

The parameters that are obtained by modelling an image region as a Markov/Gibbs Random Field are representatives of the variation (texture) of that region. Most often we assume a homogeneous and isotropic model (see section 4.4.1). Thus most pixels in that block are expected to satisfy the model and its parameters. Those that do not can be treated as outliers (noise) and can be smoothed to ensure homogeneity. The smoothing operation can be implemented as a probabilistic approach wherein the expected value and the present value are compared according to the model involved. But a more straightforward approach, that works well, is to replace the value of the central pixel by the expected value as computed from the neighborhood and model parameters. Modifications can be done in the same pass or each pass can be updated once all the sites have been processed. Figure 4.11 shows an instance of figure 4.10 with added Gaussian noise. Figures 4.12, 4.14, 4.16 and 4.18 show the two dimensional Gaussian filter (convolution mask) that are used to smoothen the noisy image in figure 4.11. The results are displayed in figures 4.13, 4.15, 4.17 and 4.19. Figure 4.20 shows a single pass of Markov Random Field based filtering perfomed on the noisy image. Figures 4.21, 4.22, 4.23 show subsequent passes.
Figure 4.10: Original image.

Figure 4.11: Image with added Gaussian noise.
Figure 4.12: 3x3 Gaussian smoothing filter with $\mu = 0$ and $\sigma = 1$

Figure 4.13: Gaussian filter in figure 4.12 applied to noisy image.
Figure 4.14: 3x3 Gaussian smoothing filter with $\mu = 0$ and $\sigma = 2$

Figure 4.15: Gaussian filter in figure 4.14 applied to noisy image.
Figure 4.16: 5x5 Gaussian smoothing filter with $\mu = 0$ and $\sigma = 1$

Figure 4.17: Gaussian filter in figure 4.16 applied to noisy image.
Figure 4.18: 5x5 Gaussian smoothing filter with $\mu = 0$ and $\sigma = 2$

Figure 4.19: Gaussian filter in figure 4.18 applied to noisy image.
Figure 4.20: MRF-based Filtering, Pass 1. (Error Norm $S_1=406.5412$)

Figure 4.21: MRF-based Filtering, Pass 2. (Error Norm $S_2=21.96$)
Figure 4.22: MRF-based Filtering, Pass 3. (Error Norm $S_3 = 10.798$)

Figure 4.23: MRF-based Filtering, Pass 4. (Error Norm $S_4 = 4.7281$)
Let $S_p$ denote the Error Norm (Standard Deviation between the pixels) of the image after pass $p - 1$ and the image after pass $p$. Pass 0 represents the original noisy image. It is interesting to note that $S_p$ decreases monotonically. Notice that these multiple passes reduce noise but do not wash out the image as is the case with the gaussian filters. Moreover, the choice of the gaussian filter is largely arbitrary or empirically determined, while the Markov Random Fields based filtering method is an adaptive process that takes into account the local properties of the image. This method also works well with speckle noise. Thus the smoothing operation using Markov Random Fields is a controlled operation as compared to other commonly used methods of smoothing.

### 4.7.2 Texture Generation

Since Markov Random Fields are used to derive a representative signature for an image (or a section thereof), it is natural to think about regeneration, i.e. the possibility of constructing a texture pattern from a representative signature. See figures 4.24 and 4.25 for examples of random texture generation from texture measures.

![Figure 4.24: Texture Generation from MRF parameter values](image)

![Figure 4.25: Texture Generation from MRF parameter values](image)
Figure 4.26: Original VISTEX texture images
4.8 Pattern Matching and Search Using Wavelet Multiresolution

When matching two regions that are modelled as a single texture, a straightforward computation of Markov Random Field parameters suffices. See figures 4.27 and 4.30. The corresponding error norm gives us a measure of the proximity of the two regions. These figures display the results obtained from an implementation using the autonormal model with least squares estimation. The leftmost column in these images display the query images. Each row displays the lowest ranked matches from the target archive, corresponding to the query which is displayed as the leftmost entry in the row. Let \( N \) be the lowest error norm and \( M \) denote the largest error norm in the target archive. If a target image corresponding to some query has an error norm of \( E \), then the percentage rank, displayed on top of each of the lowest ranked target images, is given by \( \frac{E}{M-N} \times 100 \). As before, the algorithm is tested on the VISTEX database (167 images). Various queries of different sizes (sections taken from the targets themselves) have been tested with satisfactory results. Note here that there is no training phase involved here. The results may be improved further by incorporating a training phase but that is outside the scope of the current work.
Figure 4.28: Matching Images 13 through 16
Figure 4.29: Matching Images 25 through 28
Figure 4.30: Matching Images 69 through 72
**Pattern search** is an extension of pattern matching. Given an image *Query* image, a pattern searching task is to find the occurrence of that region within another *target* image region. The target region may be smaller or larger in size than the query. In our discussion, we will assume the target region to be larger than the query. A simple approach is to apply pattern matching on *all* the blocks in the target image that are of the same size as the query. But this defeats the power of the Random Field approach, mainly the size-invariance property of the texture parameters.

A much more prominent problem is that of the computation involved. For example if the query is a $64 \times 64$ image and the target is of size $512 \times 512$, the number of blocks of the target that would have to be compared is $(512 - 64) \times (512 - 64) = 200704$. Each such block has its own set of computations to come up with the texture parameter values. In all, the computation requirement is high. An alternative would be to increase the block size for the target. But a better approach is to use a wavelet based multiresolution approach. Instead of working directly on the query and target images, one performs a wavelet decomposition on both the query and the target images. The average components from the decomposition form the new query and target respectively. Assuming a dyadic wavelet transform, each level of the transform reduces the size of the query and target by a factor of $\frac{1}{4}$. This leads to an overall reduction of computation to $\frac{1}{16}$ of the original amount. The number of levels, of the decomposition, to be used is empirically determined and it depends upon the contents and size of the images under consideration.

### 4.9 Conclusion and Future Direction

The Markov Random Field model of texture representation is a very versatile tool for image processing applications. It has a number of different uses, as described above. Different stochastic models may be applied to the Markov Random Field formulation to model the texture pattern in question. Multiresolution when combined with texture extraction, reduces processing times for texture matching applications, many fold. There is a lot of scope for improving the performance of the algorithm further. Additional models, parameter estimation techniques, training phases, etc. may be employed. The wavelet based measure extraction process may be combined with the Markov Random Field method to form a combined measure that will incorporate both positional information as well as local image properties.
Bibliography


Vita

Intiaz Hossain was born on the 18th of February, 1976 in the city of Kolkata in the eastern part of India. He completed his undergraduate degree in computer applications at the International Institute of Professional Studies, Devi Ahilya University, Indore, India in 1998. Thereafter he came to the United States of America and has pursued two masters degree programs, in Mathematics and Electrical Engineering respectively. He will be graduating with the respective degrees in the Summer and Fall respectively of 2004. His interest areas include image processing, pattern recognition and harmonic analysis.