SIGNAL PROCESSING FOR A LIDAR SYSTEM

by

DEBASISH SASMAL
B.Tech., Indian Institute of Technology, Kharagpur, India, 1998

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

We accept this thesis as conforming
to the required standard

[redacted]

Dr. A. Antoniou, Supervisor, Dept. of Elect. & Comp. Eng.

[redacted]

Dr. D. J. Shpak, Supervisor, Dept. of Elect. & Comp. Eng.

[redacted]

Dr. G. McLean, Outside Member, Dept. of Mechanical Eng.

[redacted]

Mr. Terry Curran, External Examiner, Institute of Ocean Sciences

Supervisors: Dr. A. Antoniou, Dr. D. J. Shpak

ABSTRACT

The advent of LIDAR technology has led to a range of applications in the field of topographical mapping. An airborne laser system can acquire a large volume of terrain data within a short span of time. The thesis is based on a project which involves the development of ground and canopy profiles from a terrain data acquired by a laser system mounted on an airborne sensor platform. The project, sponsored by Terra Remote Sensing Inc., also involves extracting wire hits, if any, from the terrain data and drawing catenary curves through the wire hits. While data acquired by a profiler laser system are good for extracting a 2-D topographical curve they are not suitable for obtaining a 3-D surface plot. Data acquired by a laser scanner system, on the other hand, are quite suitable for obtaining a 3-D terrain model.

The process starts by classifying the data points as ground hits, canopy hits or wire hits using a method based on order statistics filters. The non-uniformly located data points are interpolated in order to obtain smooth profiles. Two methods of interpolating non-uniform profiler data have been used. The first is a multiscale process wherein the data are regularized using an iterative piecewise polynomial method. The second method is based on maximum smoothness interpolation in the wavelet domain. The wavelet based interpolation technique has also been used to interpolate data acquired by a scanner system. A second method used for interpolating scanning data involves fitting a smooth curve in each of the triangular cells formed by drawing a triangulated network with data points as vertices of the network. Finally, catenary wires have been traced by joining the wire hits and eliminating data points which might have wrongly been classified as wire hits.

PARTIAL COPYRIGHT LICENSE

I hereby grant the right to lend my thesis to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the Library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Title of Thesis: SIGNAL PROCESSING FOR A LIDAR SYSTEM.

Author: _____

DEBASISH SASMAL
March 2001

Examiners:

[redacted]

Dr. A. Antoniou, Supervisor, Dept. of Elect. & Comp. Eng.

[redacted]

Dr. D. J. Shpak, Supervisor, Dept. of Elect. & Comp. Eng.

[redacted]

Dr. G. McLean, Outside Member, Dept. of Mechanical Eng.

[redacted]

Mr. Terry Curran, External Examiner, Institute of Ocean Sciences

Table of Contents

List of Figures

Acknowledgement

At the very outset I convey my deep respect and gratitude to my supervisors Dr. Andreas Antoniou and Dr. Dale Shpak, for their valuable guidance and comments throughout my graduate studies.

I would like to acknowledge Terra Remote Sensing Inc. for their financial support without which I would not have been able to undertake my research work during my tenure as a graduate student.

I am thankful to the computer and secretarial staff of the Department of Electrical and Computer Engineering, in particular, Vicky Smith, for the support extended to me. I would also like to thank all my friends who became an integral part of my student life and extended their help both academically and otherwise.

I would like to specially thank Sanghamitra Ghosh, who has now become a special person for me, for her help and constant encouragement during the thesis work. Finally, I would like to thank my family who encouraged me throughout my graduate studies and gave me the will to continue.

Dedication

Dedicated to my family

List of Abbreviation

AOL	Airborne oceanographic LIDAR
DD	Deslauriers-Dubuc
DWT	Discrete wavelet transform
LIDAR	Light detection and ranging
OSF	Order statistics filter
PLADS	Pulse-light airborne depth sounder
PRF	Pulse repetition frequency
TM	Topographic map

Chapter 1

Introduction

Topographic mapping has been an ever evolving process with numerous researchers actively involved in developing more efficient means of providing needed data products. The need for an ever growing volume of surveying and mapping data and higher data density warrant the ongoing research involved in the development of cost-effective methods for application to topographical mapping. This provides an impetus for developing airborne laser technology for application to surveying and mapping.

The concept of using an airborne laser system for terrain mapping evolved from oceanographic applications of light detection and ranging (LIDAR) starting in the late 1960s. First used for bathymetry in 1968, the use of airborne laser techniques was demonstrated with a system constructed at the Syracuse University Research Corporation by Hickman and Hogg [1]. In laser bathymetry a series of short, intense pulses of laser light is projected from the aircraft into the ocean. The depth of the ocean bed can be deduced from the surface reflection and the laser light reflected from the ocean bed. Backscattered radiation from the water column is also received and can be analyzed to estimate the turbidity of the sea water. The first commercially built research system was the pulse-light airborne depth sounder (PLADS) developed by Raytheon [2]. Hoge et al. reported the results of a study to derive water depths in the Atlantic Ocean and in the turbid Chesapeake Bay using a system developed for NASA, known as the airborne oceanographic LIDAR (AOL) system [3] [4].

The attributes which make an airborne laser so useful for ocean profiling have been known to be relevant to terrestrial problems. Initial interests in the terrestrial applications of airborne laser was on topographic mapping of the terrain. The ability of the laser profiler to penetrate the vegetation so that the ground elevations beneath a continuous vegetation cover could be derived was used to map the vegetation cover

of the terrain. Link and Collins [5] provided a summary of the operational aspects of a laser terrain mapping system. Krabill et al. [6] used an airborne laser equipped with a vertical accelerometer to map the topography of a watershed near Memphis, TN. Arp et al. [7] reported on an ambitious laser mapping project for water reservoir location in an inaccessible area in Venezuela. Schreier et al. [8] determined in a terrain mapping study in Canada that 95% of all laser terrain elevations were within 1.80m of photogrammetrically derived values. Vegetation cover had a significant influence on the accuracy of the laser terrain measurements; dense vegetation introduced an element of noise into terrain mapping. They noted that laser profiling data would be useful for vegetation canopy height determination. Nelson et al. [9] utilized the tree height-finding capability of a profiling laser on a heavily forested hardwood site in Pennsylvania to show the changes in the laser canopy profile corresponding to changing canopy density conditions. Schreier et al. [10] reported that their near-infrared laser produced accurate terrain and vegetation canopy profiles, thereby permitting accurate measurement of tree height. They used the canopy height measurements in conjunction with the near infrared reflectance value to tree types (conifer, hardwood) using semiautomated classification techniques. Maclean [11] demonstrated that the cross-sectional area of the forest canopy derived via photointerpretation is directly related to the natural logarithm of the gross merchantable timber volume.

In the forestry experiments accomplished so far the lasers have ususally been operated in a profiling mode, i.e., only a single line of data directly beneath the aircraft is recorded. For large scale practical applications of laser data the profiling technique provides a rather limited sample. Scanning laser systems would provide greater distributions of samples, although the consequences of looking through the forest canopy at an angle need to be better investigated [12]. Essential for the success is a high intensity of measurements, the ability to locate accurately the three-dimensional coordinates of the objects reflecting the pulsed laser beams [13] [14] [15] and narrowly focused laser beams producing well-defined return signals. Today's advanced systems meet these requirements [16] [17].

1.1 Concept and Operation

The thesis is based on a project sponsored by Terra Remote Sensing Inc. The project involves the development of an airborne scanning laser to measure distances between transmission-line wires and ground targeted from an accurately positioned airborne sensor platform. A high pulse-repetition-frequency (PRF) laser directed by a scanning mirror system will be used to capture returns from transmission wires and the terrain backgrounds. The laser scanner data will then be used to create a topographic map (TM) of the ground. Software routines will extract the wire hits from the laser data and using catenary formulae, fit a curve through the points. Wire and ground profiles can then be extracted from the data sets. The resulting profiles for the wires, vegetation and ground are used for determining safe clearances for high-voltage transmission lines. This development will build on Terra's previous work with lasers used in its LIDAR profiler system. Terra has developed infra-red/blue/green 30 Hz profiling lasers for coastal mapping applications and will use this as its building block for integrating a 2 kHz scanning laser into the proposed project.

A Bell 206B helicopter is the vehicle which has been planned for carrying the sensor package. The proposed laser range finder uses a pulsed laser diode emission in the infrared spectrum at 905 nanometers. The PRF of the laser is variable, but a PRF of 2000 Hz has been proposed. The laser scanner distributes the laser pulses in a defined pattern across the flight path of the aircraft. The scanner will consist of an oscillating mirror coupled to a shaft encoder. The pattern of oscillation will be from +15 to -15 degrees from nadir for a total scan sweep angle of 30 degrees. Figure 1.1 illustrates the geometry of the scanning laser system.

1.2 Scope of Thesis

The objectives of the thesis are to explore the use of signal processing for the following processes:

- classification of laser hits as ground hits, vegetation hits or possible transmission wire hits,
- delineation of the ground profile,

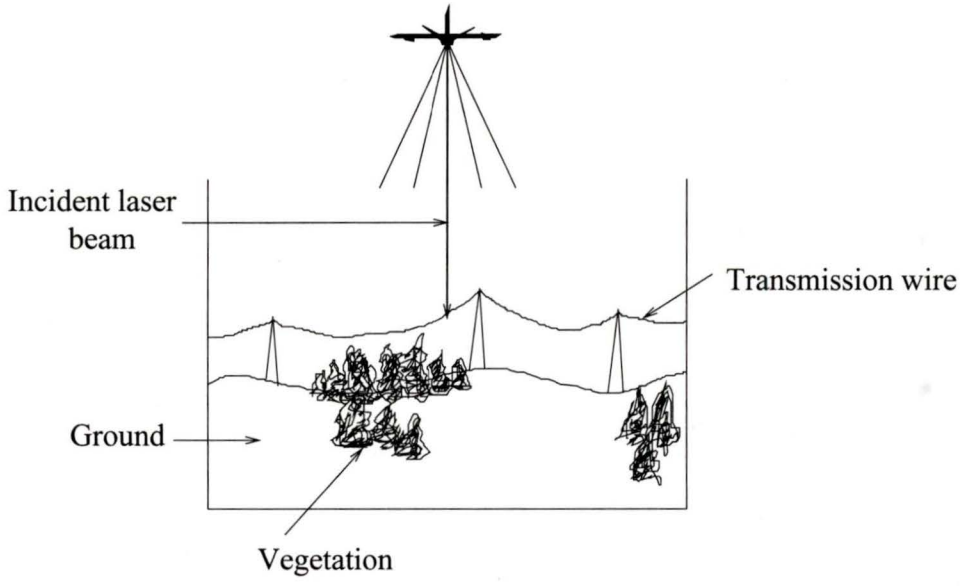


Figure 1.1. *Laser-beam geometry for the scanning laser system.*

- delineation of the canopy profile and,
- tracing of transmission wire catenaries, if any.

One-dimensional signal processing algorithms are used to process the LIDAR data aquired by an existing profiler system. These algorithms are further improved and modified to process data aquired by a scanner system by using 2-D signal processing techniques.

Chapter 2 describes a set of signal processing algorithms that can be used to pre-process the LIDAR data in order to facilitate automatic classification of data points. The data are suitably archived and data structures maintained in order to make the processes time and memory efficient. The data are normalized and translated from the x-y coordinate system to the x'-y' coordinate system where the x'-y' coordinate system is formed by rotating the x-y coordinate system by a certain angle. A searching algorithm which determines the nearest neighbors of a particular data point has been described. To facilitate the searching process the data set is broken down into smaller blocks. A method based on order statistics filters (OSF) has been developed in order to classify the data points as ground hits, vegetation hits or possible transmission wire hits. Vegetation-top hits are further identified from the set of vegetation hits.

The data points classified as ground hits and vegetation-top hits are nonuniformly scattered and thus, to obtain a 2-D topographical curve for a profiler data and a 3-D surface plot for scanner data, nonuniform interpolation techniques are used to interpolate data onto a regular grid. Chapter 3 discusses two methods of 1-D nonuniform interpolation. The first is a multiscale iterative interpolation technique based on the Deslauriers-Dubuc process and the second is based on maximum smoothness interpolation in the wavelet domain. The iterative approach tries to fill-in *missing* data points on a regular grid by using a piecewise polynomial method. On the other hand, the wavelet-based interpolation process is considered as an optimization problem where the original signal is estimated from the given signal samples according to some optimality criterion. The behavior of the two processes is discussed and their performances compared.

Chapter 4 discusses two processes for interpolating data points scattered on a 2-D profile. The first is a triangle-based interpolation process wherein a triangulated network is constructed with the data points forming the vertices of the triangulated network. The resulting surface is modeled as a combination of surfaces, each surface modeled over a triangular cell in the network. A fifth-degree polynomial is fitted in each triangular cell. The second method is an extension to two dimensions of the wavelet-based process discussed in chapter 3. The formation of the wavelet transform matrix, which converts a 2-D signal to its wavelet coefficients, is discussed in detail.

Transmission wires, if any, are traced considering the fact that the projection of the wires on the x-y plane form straight lines. Actual wire hits are identified from the set of possible wire hits. An optimization method of fitting a smooth curve through the points using catenary formulae has been presented. Chapter 5 discusses the algorithms developed to obtain the wire profiles.

Conclusions and directions for further research are found in chapter 6.

Chapter 2

Preprocessing of LIDAR Data

2.1 Introduction

The LIDAR data consist of the x and y coordinates and the corresponding elevation values obtained from the laser reflections. Depending on the topography of the region laser beams can be reflected off the ground, bushes, trees, buildings, cars, transmission wires, or even from flying birds. The data can also have occasional impulsive noise due to errors in the data acquisition system or for some other reason. In order to obtain the ground and canopy profiles of the region from where the data have been acquired and also to delineate transmission wires, if any, the data points will first have to be classified as ground hits, vegetation hits, wire hits or anything other. Section 2.5 discusses in detail the classification of the laser hits.

Once the data points have been classified, data is interpolated on a regular grid to obtain the profiles. Transmission wires, if any, are delineated considering the fact that wires maintain catenary curves from pole to pole. These procedures will be discussed in subsequent chapters. In order to make the procedures time and memory efficient, various data structures will be maintained while archiving the data and also during the implementation of the procedures.

2.2 Data Archiving

In order to make the processing more time and space efficient, it is necessary to archive the data. For this reason some data-structures are maintained and the data points suitably grouped.

2.2.1 Rotation and normalization of the data

The direction of the flightline of the aircraft which acquires the data depends on the topographical pattern of the region. When the aircraft maintains a particular flightline the sounding pattern of the data generated on the surface has an orientation which is related to the flightpath. More often the flightline is not parallel to the x axis, the data points being represented by x (easting) and y (northing) coordinates. Hence, for a convenient way to process the 2-D data, the data is first transformed to a new coordinate system, the x'-y' system.

The data point (x_i, y_i) is transformed to (x'_i, y'_i) according to the relations

$$\begin{aligned} x'_i &= r\cos\{\tan^{-1}(y_i/x_i) - \theta\} \\ y'_i &= r\sin\{\tan^{-1}(y_i/x_i) - \theta\} \end{aligned} \tag{2.1}$$

where

$$r = \sqrt{{x_i}^2 + {y_i}^2}$$

and θ is the angle by which the x-y coordinate system is rotated to form the x'-y' coordinate system.

The data points are then normalized to the values $(\hat{x}, \hat{y})$ according to

$$\begin{aligned} \hat{x}_i &= (x'_i - xmin)/(maxmin) \\ \hat{y}_i &= (y'_i - ymin)/(maxmin) \end{aligned} \tag{2.2}$$

where

$$maxmin = maximum(xmax - xmin, ymax - ymin)$$

and

$$\begin{aligned} xmax &= maximum\{x'_i\} \\ xmin &= minimum\{x'_i\} \\ ymax &= maximum\{y'_i\} \\ ymin &= minimum\{y'_i\} \end{aligned}$$

This ensures that the values of $\hat{x}$ and $\hat{y}$ lie between 0 and 1.

2.2.2 Grouping of data points

A lot of searching is involved during the classification of data points and also while data are interpolated. In order to make the searching process more efficient, the data set is divided into smaller blocks. Thus, to search for data points in the neighborhood of a particular data point located in a particular block it is sufficient to search for data points in that particular block and, if need be, in adjacent blocks instead of searching the whole data set. This considerably reduces the search time. The next section describes a simple algorithm to search for data points in the neighborhood of a particular data point. Here the focus is to determine the local characteristics of the surface at the location of the particular point using a few data points near it and so importance is given not on the accuracy of the searching algorithm to find the nearest neighbors but on the simplicity and efficiency of the data structures that would help in further processing of the data. Other similar hierarchical searching schemes can be used based on quad-trees [18], [19].

The size of the blocks may be appropriately chosen depending on the density of the data points. The greater the number of data points in a block, the greater is the number of data points which have to be searched in order to determine the nearest neighbors. So the size of the blocks is so chosen so that each block contains a few data points.

Once the sizes of the blocks have been finalized, data points belonging to the corresponding blocks are grouped. A linked list is maintained for each block, the nodes of the linked list being the data points included in the block. Each node in the list consists of the x and y coordinates of the data point it represents and also its elevation value. A separate field in the block keeps track of the number of data points included in the block. Memory is allocated dynamically depending upon the number of data points included in the block.

2.3 Search for Neighboring Points

Let the desired number of neighbors for a particular point p_0 located in block $b_{i,j}$ be n. The search to find n neighbors close to p_0 starts by finding the number of data points contained in block $b_{i,j}$. If $b_{i,j}$ contains more than n data points, there may not

be a need to search for data points located outside $b_{i,j}$. If the number of data points contained in $b_{i,j}$ is less than n, all of the blocks adjacent to $b_{i,j}$ are also searched. If the total number of data points included in $b_{i,j}$ and all its adjacent blocks is still less than n, blocks adjacent to these blocks are also searched and so on.

It is always possible that for p_0 to be located near the boundary of $b_{i,j}$ and $b_{i,j}$ to contain more than n data points. In such a case, the nearest neighbors of p_0 will be selected among the data points included in $b_{i,j}$ itself although other data points included in blocks adjacent to $b_{i,j}$ may be nearer to p_0 as illustrated in Figure 2.1.

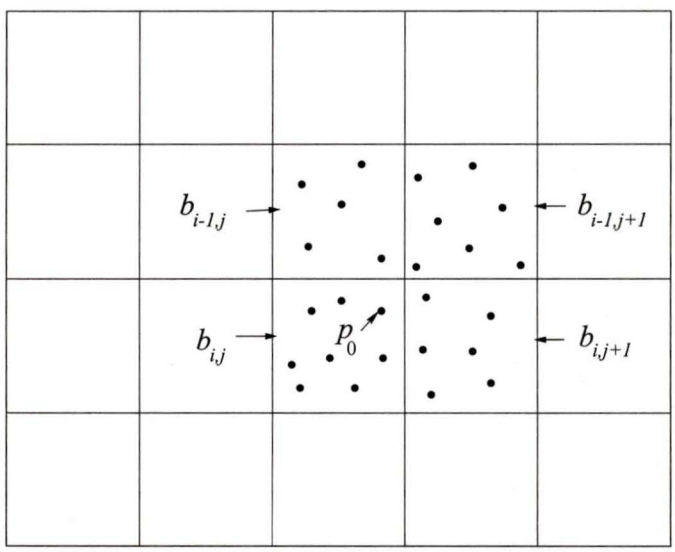


Figure 2.1. *Point p_0 is nearer to some points in adjacent blocks than some of the points in block $b_{i,j}$ itself.*

In order to avoid false identification of nearest neighbors, data points located in adjacent blocks are also included in the search to find the n nearest neighbors of p_0. For this, a simple criterion has been used. Block $b_{i,j}$ is broken into five parts: *center*, *top-left*, *top-right*, *bottom-left* and *bottom-right* as illustrated in Figure 2.2. If p_0 is located inside the *top-right* part of block $b_{i,j}$ data points contained in adjacent blocks $b_{i-1,j}, b_{i-1,j+1}$ and $b_{i,j+1}$ are also considered. Similarly, if p_i is located in any of the other three parts of block $b_{i,j}$ denoted by the shaded regions in Figure 2.2, then blocks adjacent to the shaded portion are also considered. The area of the shaded parts can be appropriately chosen. If p_0 is located inside the *center* part of block $b_{i,j}$ then no

adjacent blocks are considered unless $b_{i,j}$ has fewer than n points.

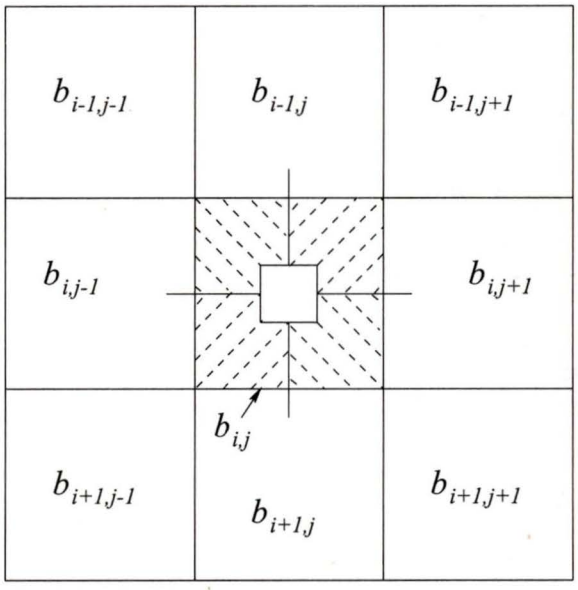


Figure 2.2. *Block $b_{i,j}$ divided into five parts.*

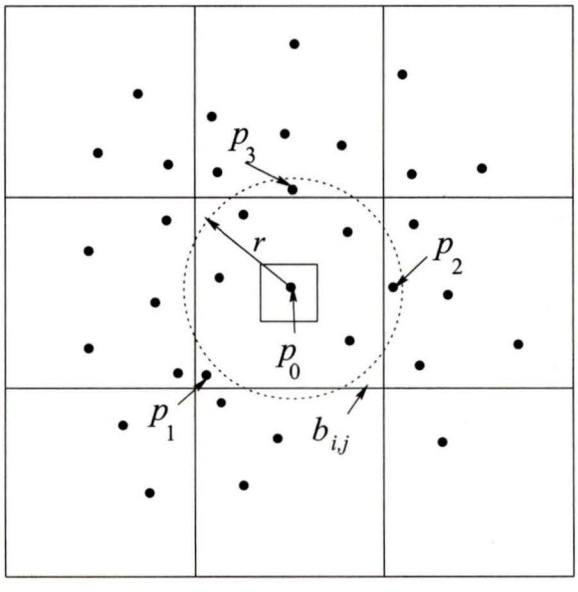


Figure 2.3. *Point p_0 nearer to points p_2 and p_3 than point p_1.*

When p_0 is located in the *center*, for some cases it may so happen that some data points in adjacent blocks are nearer to p_0 than some of the data points selected

as its nearest neighbors from within block $b_{i,j}$ itself. We see in Figure 2.3 that p_1 is considered as one of the $n = 5$ nearest neighbors of p_0 although p_2 and p_3 located in adjacent blocks are nearer to it. Thus, it is advisable to search for m data points, where $m > n$, and then select the n points nearest to p_i as its nearest neighbors. Although this may eliminate the problem of false identification of the nearest neighbors to some extent, for some cases when most of the data points in $b_{i,j}$ are concentrated at the corners, this may still not work. Since data points are usually approximately evenly distributed, it is not expected that data points in any block will be concentrated only at the corners.

2.4 The Order Statistics Filter

In order to classify data points as ground hits, vegetation hits or transmission wire hits, a method based on order-statistics filters (OSF) has been developed.

If y_k is the output of the order-statistics filter with window size $2N + 1$, then for an input sequence x_j, $j = 1, 2, \ldots, 2N + 1$, y_k is given by

$$y_k = \sum_{j=1}^{2N+1} w_j x_j \tag{2.3}$$

where x_j is the jth smallest sample among the $2N + 1$ samples inside the window centered at k and w_j is a constant weight applied to x_j. Note that when $w_{N+1} = 1$ and the rest of the weights are set to zero, the OSF becomes a median filter; when $w_{q+1} = w_{q+2} = \cdots = w_{2N+1} = 1/[2(N - q) + 1]$ where $0 \leq q \leq N$ and the rest of the weights are set to zero, the OSF becomes an alpha-trimmed mean filter. Thus, by appropriately selecting the weighting coefficients, we can use the OSF for specific needs, and in particular to classify height estimates scattered on a 2-D domain.

2.5 Classification of Data Points

Suppose we have a set of data points d_i, $i = 1, 2, \ldots, N$ scattered on a 2-D domain. We define a neighborhood of d_i and search for its m closest height estimates. The spatial characteristics of the data samples in the neighborhood are then used to classify d_i. The data points in the neighborhood are initially arranged in ascending

order of their elevation values, such that $h_1 \leq h_2 \leq \cdots \leq h_m$, where h_i represents the ith element in the ordered array. The data point having the least elevation value, h_1, can be considered to be a ground hit assuming that there are no data points with an impulsive elevation value lower than the actual ground elevation. The point d_i is classified as a ground hit if its elevation value does not exceed that of h_1 by a certain threshold value, i.e.,

$$elevation\ (d_i^g) \ \leq \ h_1 \ + \ th_g \tag{2.4}$$

for a suitably chosen threshold th_g. Similarly, d_i is classified as a vegetation hit if the elevation value of d_i exceeds h_1 by some threshold th_{vl} but does not exceed by a certain threshold th_{vh}, for $th_{vl} < th_{vh}$, i.e.,

$$h_1 \ + \ th_{vl} \ \leq \ elevation\ (d_i^v) \ \leq \ h_1 \ + \ th_{vh} \tag{2.5}$$

Considering the fact that transmission wires maintain a minimum elevation from the ground and also pass above the vegetation in that region, d_i can be accordingly classified as a transmission wire hit if its elevation value exceeds h_1 by a certain threshold th_w, i.e.,

$$elevation\ (d_i^w) \ \geq \ h_1 \ + \ th_w \tag{2.6}$$

Note that this scheme will result in false categorization of d_i if the neighborhood of d_i contains even a single impulsive elevation value which is much less than the actual ground value at that point. Moreover, the scheme does not distinguish between a data point which is near d_i and one which is far from it. Since, in general, the spatial correlation of a typical topographical datum decreases with increasing distance from the point under consideration, undue influence of distant data points should be avoided. Therefore, instead of considering just the first data point in the ordered array of data points to compare with d_i it is advisable to discard the first few data points in the ordered array and consider more than one point as ground hits. An array of height estimates X can be formed as

$$X \ = \ \{h_{f+1}, h_{f+2}, \cdots, h_{f+g}\}$$

where the number of discarded data points is f and the number of data points considered to be ground hits is g. By applying a suitable weighting function over X one

can avoid the undue influence of distant points. One such weighting function is the inverse distance-powered function given by

$$w_j = \frac{A}{{l_j}^b}$$

where l_j represents the Euclidean distance between d_i and the sounding location of h_j. The parameter b controls the degree of influence of distance on the prediction, and A is a normalizing constant defined as

$$A = 1/\left(\sum_{j=f+1}^{f+g} \frac{1}{{l_j}^b}\right)$$

such that

$$\sum_{j=f+1}^{f+g} w_j = 1$$

Thus by applying w_j to h_j in X, the ground elevation value at the location of d_i can be predicted. Let $\hat{h}_i$ denote the predicted ground elevation value at the location of d_i. Replacing $\hat{h}_i$ for h_1 in (2.4), (2.5) and (2.6), d_i can be classified as a ground hit, vegetation hit or a wire hit, respectively.

Note that at places where there is no vegetation the canopy profile should coincide with the ground profile. To ensure that this is the case, the lower vegetation threshold value th_{vl} is kept low so that ground hits are also classified as vegetation hits at these regions.

Data points classified as vegetation hits may include laser hits reflected from the top of trees, bushes, etc., as well as hits which penetrate the vegetation. Thus to get the canopy profile only the vegetation-top hits should be selected. To categorize vegetation-top hits from vegetation hits classified earlier, a similar approach is used as was done to classify data points. For each vegetation hit v_i, we define a neighborhood of v_i and search for its m closest neighbors. The data point having the highest elevation value among all the data points within the neighborhood is considered as a vegetation-top hit and is grouped in a separate set.

Data points classified as transmission wire hits, if any, may include laser hits reflected from birds, poles supporting the wires and also may include impulsive noise values. Hence, to delineate transmission wires, data points classified as wire hits are

further processed to determine actual wire hits by using the fact that the projection of the transmission wires onto the x-y plane maintains straight lines. This is discussed in detail in chapter 5.

Thus, with the data points accordingly grouped as ground hits, vegetation-top hits or transmission wire hits, the stage is now set to obtain the ground and canopy profiles and to trace the transmission wires, if any.

Although ground hits, vegetation hits and possible transmission-wire hits are grouped in seperate sets, the data points are not actually reproduced. In order to save memory space, the nodes representing the grouped data points are declared as pointers and are not allocated any memory. These pointers merely point to the location where the data points were located in the original data set. The grouped data points thus form virtual data sets consisting of some points selected from the original data set itself. From the virtual set consisting of possible wire hits selected points are considered as actual wire hits. These data points in turn form a virtual set.

2.6 Conclusions

For convenient processing of LIDAR data, the data has been initially translated to another coordinate system, the x'-y' system and then normalized. The data set is further broken down into several blocks. The data points are spatially grouped into appropriate blocks. This effectively helps in the search for the nearest neighbors of a particular point and makes the process more time-efficient.

Data points have been classified as ground hits, vegetation hits or transmission-wire hits using a method based on an order-statistics filter. Data points are accordingly classified by comparing their elevation values with the elevation value of an assumed ground hit in their neighborhood. Vegetation-top hits are then identified from the set of vegetation hits using a similar approach. Suitable data structures have been maintained so that the memory requirement is minimized.

Chapter 3

Signal Processing for a LIDAR Profiler System

3.1 Introduction

The data aquired by a LIDAR profiler system is nonuniform and so the data needs to be interpolated in order to get the ground and vegetation-top profiles. Thus, we have a nonuniform interpolation problem. Nonuniform interpolation over scattered data refers to the problem of fitting a smooth function through a set of nonuniformly distributed data samples. Despite a lot of activity in this area, nonuniform interpolation over scattered data remains to be a difficult and computationally extensive problem. There is a vast amount of literature devoted to nonuniform interpolation which documents various approaches, many of which suffer from limitations in smoothness, time complexity, or allowable data distributions [20].

One of the earliest methods used was based on inverse distance weighting of data which became known as Shepard's method [21]. Shepard defined a C^0 continuous interpolation function, a function which has no continuous derivatives, as the weighted average of the data, with the weights being inversely proportional to the distance. This method has several shortcomings. It is a global method and requires all the weights to be recomputed if any data point is added, removed or modified. Furthermore, it suffers from undue influence from points which are far away, and hence the interpolated data is not representative of local characteristics of the data. Franke and Nielson introduced the modified quadratic Shepard's method [22] to address these deficiencies and to produce C^1 continuous interpolation functions which have only first-order continuous derivatives.

Another popular approach to interpolation, and by far the most common technique, is polynomial interpolation wherein a polynomial, of certain degree, is fitted over the data samples. However, the use of polynomial interpolation offen gives unsatisfactory results when used for practical problems. In order to use polynomial interpolation for practical problems it may be neccessary to use polynomials of a very high degree. As a polynomial with a high degree is very oscillatory in behavior it is unsuitable for interpolating functions which are smooth in nature.

An alternative approach is to define the interpolating function as a combination of lower-degree polynomials. These polynomials are joined together to form a piecewise polynomial which is then used to interpolate the data. However, the interpolated function often shows oscillatory behavior and thus is unsuitable for problems where the functions are fairly smooth in nature. The resulting curves may also have discontinuous derivatives if continuity conditions are not imposed between successive polynomials.

Another popular approach is to define the interpolation function as a linear combination of basis functions or splines. This method has the property that it interpolates at each of the data points and smoothes out as much as possible between the points. Thin plate splines, derived by minimizing the integral of the curvatures over the domain among the interpolation functions of the data points, are widely used.

Another class of solutions is due to finite element methods. One approach, mainly used for 2-D nonuniform interpolation, involves forming a triangulation network, with the data points as vertices of the triangles, over which surface patches are fitted. This approach will be discussed in more detail in the following chapter.

Schumaker [23] proposed a two-stage method that first generates a grid of data using any suitable method for nonuniform interpolation and then applies a standard tensor product approximation on the grid. The resulting approximation may be made to interpolate the original data through the use of an iterative process called the delta iteration, developed by Foley and Nielson [24].

Deslauriers and Dubuc presented a multiscale method of interpolating uniformly sampled discrete data points using an iterative refinement technique [25] [26]. Given a set of uniformly sampled data points the refinement technique finds the interpolating values of the function at points halfway between the given points. This method will

be discussed in further detail in section 3.2. Based on this technique a simple method for interpolating nonuniformly sampled data points will be presented in section 3.3.

A new approach to signal interpolation from nonuniform samples, formulated by Choi and Baraniuk [27], is the multiscale maximum-smoothness interpolation. In the maximum-smoothness interpolation approach a signal is selected passing through the samples which has the minimum smoothness norm in an appropriate function space. This technique will be discussed in detail in sections 3.4-3.6. Choi and Baraniuk have shown that the maximum interpolation technique [27] yields better interpolation results when compared to the optimal bandlimited interpolation and cubic spline interpolation techniques.

The 1-D interpolation techniques will be used for interpolating profiler data. The wavelet based interpolation technique will be introduced for interpolating 1-D nonuniform data and will be extended to interpolate 2-D nonuniform data in the next chapter.

3.2 The Deslauriers-Dubuc Interpolation Process

Definition: A point (on the variable axis) is said to be a binary rational at scale j if it can be expressed as $\frac{k}{2^j}$ for some arbitrary integers j and k.

The Deslauriers-Dubuc (DD) interpolation is a multiscale refinement technique for interpolating uniformly sampled discrete data points. In a multiscale interpolation process the set $\{t_k, k \in Z\}$ is approximated by a set of binary rationals $\{\hat{t}_k, k \in Z\}$. The approximation is justified by two reasons. Firstly, for an arbitrary t_k and a given tolerance ϵ, there always exists a binary rational $\hat{t}_k$ such that $|t_k - \hat{t}_k| < \epsilon$, and secondly, for a smooth $f(t)$, $|f(t_k) - f(\hat{t}_k)|$ can be made arbitrarily small with a binary rational $\hat{t}_k$.

Given a set of data points $\{v_k = f(\hat{t}_k)\}$ at binary rationals $\hat{t}_k = \frac{k}{2^j}$, $j \geq 0$, $k_1 \leq k \leq k_2$ for $j,\ k \in Z$, the DD refinement technique tries to find the interpolating values of the function f at all binary rationals $\hat{t}_{k'} = (k + \frac{1}{2})/2^j$, $j \geq 0,\ k_1 \leq k < k_2$ for $j,\ k \in Z$. In other words, it finds the values of $\tilde{f}(t)$ at all points halfway between previously defined points. Let $L \geq 0$ be an integer and D be defined as $D = 2L + 1$. The DD refinement process can be described as follows:

- $D+1$ values of $\tilde{f}(t)$ at $\frac{k-L}{2^j}$, $\frac{k-L+1}{2^j}$, $\ldots$, $\frac{k}{2^j}$, $\ldots$, $\frac{k+L+1}{2^j}$ are used to uniquely determine a polynomial of order D, which is denoted by $P_{j,k}(t)$, such that the values of $P_{j,k}(t)$ and $\tilde{f}(t)$ at the above points are identical.
- $\tilde{f}[(k+\frac{1}{2})/2^j] = P_{j,k}[(k+\frac{1}{2})/2^j]$
- The above two steps are repeated for $k_1 \leq k \leq k_2,\ k \in Z$.
- The above three steps are repeated for the next higher scale $j+1$.
- The above four steps are repeated until $\tilde{f}(t)$ is defined at a prescribed scale J.

There are a number of desirable features that the DD refinement process enjoys:

1. For $D = 1$ (i.e., $L = 0$), the DD process is just linear interpolation.
2. For $D = 3,\ 5,\ 7,\ \ldots$ it can be shown that this scheme defines a function which is uniformly continuous at the rationals and hence has a unique continuous extension to the reals.
3. If we start with a Kronecker sequence, which is defined as

$$v_k = \begin{cases} 1 & k = 0 \\ 0 & k = \pm 1,\ \pm 2,\ \ldots \end{cases}$$

then the DD refinement process generates interpolating functions. Figure 3.1 shows the interpolating functions for corresponding values of D.

3.3 An Iterative Approach to Nonuniform Interpolation

The problem of signal interpolation from nonuniformly sampled data can be stated as follows. Let $f(t)$ be the signal under consideration. Here, we assume that $f(t)$ is defined on the interval $I = [0, 1]$. In, addition, let $\{t_1, \ldots, t_N\}$ be the sampling points on the interval, where N is the number of samples. Without loss of generality we assume that $0 \leq t_1 < t_2 < \cdots < t_N \leq 1$. The available data are the sample values at the sampling points, that is, $\{f(t_1),\ \ldots,\ f(t_N)\}$.

Interpolation of discrete data points is the problem of finding a continuous function $\tilde{f}(t)$, for $t \in I$, having a certain degree of smoothness such that $\tilde{f}(t_k) = f(t_k),\ k = 1,\ 2,\ \ldots,\ N$

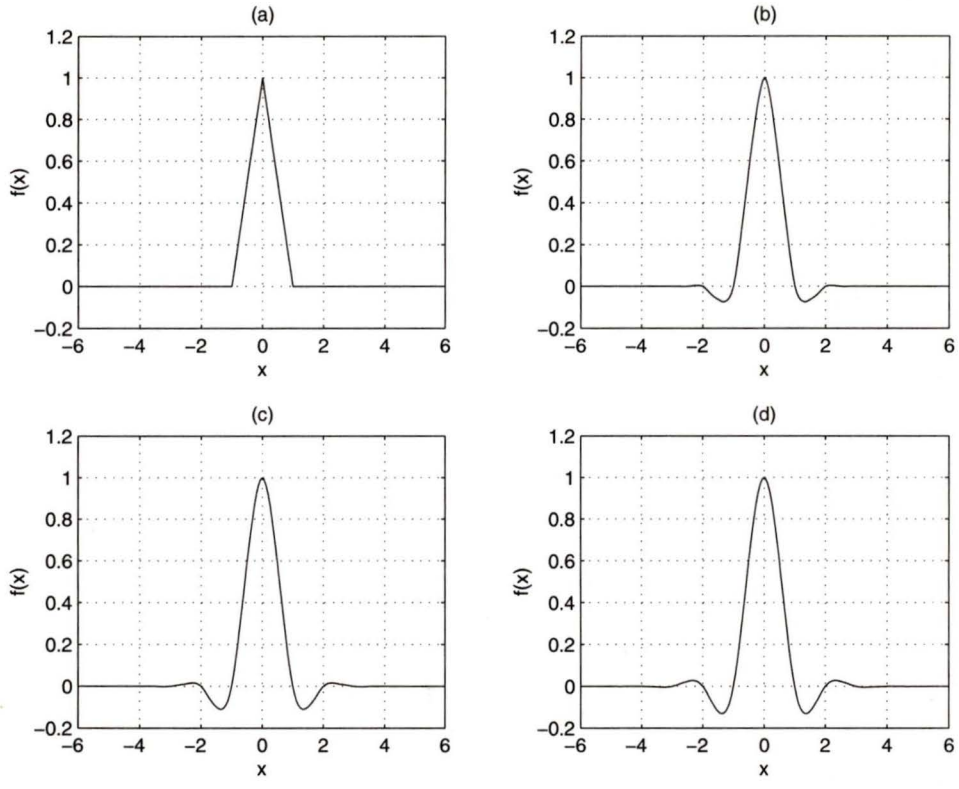


Figure 3.1. *Interpolating functions generated from the Kronecker sequence for* (a) $D = 1$, (b) $D = 3$, (c) $D = 5$ *and* (d) $D = 7$.

The multiscale point of view in the DD interpolation process can be extended to interpolate a set of nonuniform data points. The method can be better explained by regarding it as a two-stage process. At the first stage, one tries to regularize the data by filling in the missing data points using a multiscale technique. With all the missing data points filled in, we have a uniform interpolation problem and the second stage of the interpolation can be carried out using the DD refinement process.

We approximate a given set of data points $\{v_k = f(t_k)\}$ by the set $\{v_k = f(\hat{t}_k)\}$, where $\hat{t}_k$, expressed as $\hat{t}_k = \frac{l}{2^j}$, $j \geq 0$ for some arbitrary integers l and j, are approximated at a certain scale $J \geq 0$. Without loss of generality, we assume $\cdots < \hat{t}_k < \hat{t}_{k+1} < \hat{t}_{k+2} < \cdots$. Consequently, we can write $\hat{t}_k = \frac{i(k)}{2^J}$ where $i(k)$ is an integer dependent on k and $\cdots < i(k) < i(k+1) < i(k+2) < \cdots$. Thus the interpolation problem can be stated as: given the point values $\{v_k = f[i(k)/2^J]\}$,

find a continuous time function $\tilde{f}(t)$ with certain degree of smoothness such that $\tilde{f}[i(k)/2^J] = v_k$ for $k \in Z$.

The first stage can be described as follows:

(a) An attempt is made at the highest scale J to fill in as many missing data points as possible, using a $D = 2L + 1$ polynomial interpolation process described in the DD process. The process starts with identifying a missing data point, say located at $\hat{t}_{k'} = \frac{k+1/2}{2^{J-1}}$ for some arbitrary integer k. Valid data points at $\frac{k-L}{2^{J-1}}, \frac{k-L+1}{2^{J-1}}, \ldots, \frac{k}{2^{J-1}}, \ldots, \frac{k+L+1}{2^{J-1}}$ are searched for and, if these $D+1$ points exist, a polynomial of order D is fitted using the data values at these locations. However, if data is missing at any of these $D+1$ points, the interpolation cannot be carried out at $\frac{k+1/2}{2^{J-1}}$ and so, the next missing data point is identified.

(b) The process is repeated until all the missing data points have been accounted for.

(c) Next, if there are still missing data points which could not be interpolated earlier, the process is repeated for a coarser scale $J-1$.

(d) If necessary the processes (a) and (b) are repeated for a further coarser scale, and so on until a desired coarsest scale, say M is reached.

(e) There may be a possibility that some missing data points are still left. If this is the case, steps (a) to (d) are repeated.

(f) If still some missing data points are left out, the coarsest scale, M is decreased to $M-1$.

(g) Steps (a) to (f) are repeated until all the missing data points have been interpolated.

When all the missing data points are filled in we have a uniform interpolation problem and so, at stage two, the DD refinement process is carried out for a predefined number of iterations.

3.4 Wavelets and Smoothness Spaces

The discrete wavelet transform (DWT) represents a 1-D, continuous-time signal f in terms of shifted and dilated versions of a lowpass scaling function ϕ given by

$$\phi(t) = \sqrt{2} \sum_{k=-\infty}^{\infty} h_k^0 \, \phi(2t-k)$$

and shifted and dilated versions of a prototype bandpass wavelet function ψ given by

$$\psi(t) = \sqrt{2} \sum_{k=-\infty}^{\infty} h_k^1 \, \phi(2t-k)$$

where the sequences $\{h_k^0,\ k \in Z\}$ and $\{h_k^1,\ k \in Z\}$ can be interpreted as the impulse response of a lowpass and a bandpass filter respectively [28] [29].

For special choices of ϕ and ψ, the functions $\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k)$, $\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k)$, $j,\ k \in Z$ form an orthonormal basis for L_2, i.e., for any function $f(t) \in L_2$, $f(t)$ can be expressed as

$$f(t) = \sum_k u_{j_0,k}\phi_{j_0,k}(t) + \sum_{j=j_0}^{\infty} \sum_k w_{j,k}\psi_{j,k}(t) \tag{3.1}$$

with

$$u_{j,k} = \int f(t)\phi_{j,k}^*(t)dt$$
$$w_{j,k} = \int f(t)\psi_{j,k}^*(t)dt$$

The scale j_0 represents the coarsest scale under consideration; the $u_{j_0,k}$'s correspond to local means at this scale [27].

Wavelets provide a simple characterization for a wide variety of function *smoothness spaces* [30]. The norms of these spaces measure the signal smoothness: smaller norms imply smoother functions. The scale of *Besov spaces* $B_q^\alpha[L_p(I)]$, $0 < \alpha < \infty$, $0 < p < \infty$, $0 < q < \infty$, are particularly useful, for they contain many life-like signals. The Besov norm $\| f \|_{B_q^\alpha[L_p(I)]}$ can be defined as a sequence norm on the wavelet coefficients of f

$$\| f \|_{B_q^\alpha[L_p(I)]} = \| u_{j_0,k} \|_p^p + \left[\sum_{j \geq j_0} \left(\sum_k 2^{\alpha j p} 2^{j(p-2)} |w_{j,k}|^p \right)^{q/p} \right]^{1/q}$$

The three parameters have natural interpretations: a p-norm of the wavelet coefficients is taken within each scale j, a q-norm is taken across scales, and the smoothness parameter α controls the rate of decay of the $w_{j,k}$ across scales. The larger the α, the smoother the function in $B_q^\alpha[L_p(I)]$ [30]. A simple but useful set of Besov spaces are the *Sobolev spaces*, obtained as $W^\alpha[L_2(I)] \equiv B_2^\alpha[L_2(I)]$ with $p = q = 2$. In the wavelet domain we have

$$\| f \|_{W_2^\alpha[L_2(I)]} = \|u_{j_0,k}\|_2^2 + \left(\sum_{j \geq j_0,k} 2^{\alpha j} |w_{j,k}|^2 \right)^{1/2}$$

3.5 Wavelet Approach to Nonuniform Interpolation

The problem is to estimate the original continuous-time signal f from the given signal samples according to some optimality criterion. Here the optimality criterion will be the Sobolev norm of the reconstructed signal, which we will minimize in order to maximize the smoothness of the reconstructed signal.

For the sake of simplicity we will approximate the set of sampled points with a subset of dyadic points at a certain scale $j = J$. The error resulting from this approximation depends on the regularity of the signal and the scale J.

We define $\mathbf{f}$ as the column vector of length $M = 2^J$ whose elements are the dyadic samples $f_i = f(i/2^J),\ i = 0,\ \ldots,\ M-1$ at the starting(finest) scale $J > j_0$. That is, $\mathbf{f} = [f_0\ f_1\ \cdots\ f_{M-1}]^T$. For finite, discrete data, an orthonormal discrete wavelet transformation matrix $\mathbf{W}$ that takes the data vector to a wavelet/scaling coefficient vector can be constructed. This will be described in further detail in section 3.6. Since $\mathbf{W}$ is orthonormal, we have $\mathbf{W}^{-1} = \mathbf{W}^T$. In general, the elements of the vector $\mathbf{Wf}$ will not equal the wavelet coefficients of the continuous time-signal $f(t)$ unless the signal samples are *prefiltered* [31]. Denoting the prefiltering matrix by $\mathbf{P}$, the DWT coefficients for scale $j < J$ can be written as $\mathbf{w} = \mathbf{WPf}$. Conversely, the correct continuous-time signal samples can be obtained from the inverse wavelet transformation by *postfiltering*. Denoting the postfilter matrix as $\mathbf{F}$, we have $\mathbf{f} = \mathbf{FW}^T\mathbf{w}$. Let $\mathbf{V} = \mathbf{FW}^T$.

Denoting the ith row of the matrix $\mathbf{V}$ as $\mathbf{v}_i^T$ and collecting the rows of the equation

$\mathbf{f} = \mathbf{V}\mathbf{w}$ only for those indices for which f_i is known, we can write the following constraint equation on the wavelet coefficients $\mathbf{w}$

$$\mathbf{S}\mathbf{w} = \mathbf{f}^N \tag{3.2}$$

with $\mathbf{S} = [\mathbf{v}_{n_1}^T \ \mathbf{v}_{n_2}^T \ \cdots \ \mathbf{v}_{n_N}^T]^T$ and $\mathbf{f}^N = [f_{n_1} \ f_{n_2} \ \cdots \ f_{n_N}]^T$.

As (3.2) is an under-determined system of equations, there exist many different solutions for $\mathbf{w}$ that match the given sampled data $\mathbf{f}^N$.

3.5.1 Maximum-smoothness interpolation

For the minimum Besov solution, $\mathbf{w}$ can be chosen such that it satisfies (3.2) and also minimizes $\|f\|_{B_q^\alpha[L_p(I)]}$. Due to the simple characterization of Besov norms in terms of the wavelet coefficients of f, we can repose this interpolation problem as a wavelet domain optimization problem. For a general Besov space $B_q^\alpha[L_p(I)]$, the problem of finding the signal that obeys (3.2) while minimizing $\|f\|_{B_q^\alpha[L_p(I)]}$ becomes a nonlinear constrained optimization problem. For Sobolev spaces, however, the solution is simpler.

3.5.2 Interpolation in Sobolev space

In Sobolev spaces, the maximum-smoothness interpolation reduces to simple weighted least-squares optimization in the wavelet domain. Let $\mathbf{w} = [w_1 \ w_2 \ \cdots \ w_M]^T$ be the wavelet coefficients of the samples and let j_i, $i = 1, 2, \ldots, M$ denote the scale of the wavelet coefficients w_i, $i = 1, 2, \ldots, M$. Denoting the ith column of the matrix $\mathbf{S}$ by s_i, we obtain the following equation by weighting the columns of $\mathbf{S}$ and the corresponding elements of $\mathbf{w}$:

$$\begin{bmatrix} 2^{-\alpha j_1}\mathbf{s}_1 & 2^{-\alpha j_2}\mathbf{s}_2 & \cdots & 2^{-\alpha j_M}\mathbf{s}_M \end{bmatrix} \begin{bmatrix} 2^{\alpha j_1} w_1 \\ 2^{\alpha j_2} w_2 \\ \vdots \\ 2^{\alpha j_M} w_M \end{bmatrix} = \mathbf{f}^N \tag{3.3}$$

Let $\tilde{\mathbf{S}} = [2^{-\alpha j_1}\mathbf{s}_1 \ 2^{-\alpha j_2}\mathbf{s}_2 \ \cdots \ 2^{-\alpha j_M}\mathbf{s}_M]$. Then, the least-squares solution of (3.3) can be written as $\tilde{\mathbf{S}}^T(\tilde{\mathbf{S}}\tilde{\mathbf{S}}^T)^{-1}\mathbf{f}^N$, and the solution to the original problem is obtained

by restoring the weighting:

$$\hat{\mathbf{w}} = \text{diag}\{2^{-\alpha j_1}, \ldots, 2^{-\alpha j_M}\}\, \tilde{\mathbf{S}}^T (\tilde{\mathbf{S}}\tilde{\mathbf{S}}^T)^{-1} \mathbf{f}^N \tag{3.4}$$

The desired time-domain interpolation can be obtained by using the inverse wavelet transform of $\hat{\mathbf{w}}$. Ignoring the error in dyadic approximation, the interpolated signal is maximally smooth in the sense of Sobolev norm among all the functions passing through the samples.

3.6 Octave-Band Filter Bank and the Discrete-Time Wavelet Transform

We consider the filter bank given in Figure 3.2. This structure implements an orthonormal discrete-time wavelet transform, if the two-channel filter bank is orthonormal. The basis functions of the discrete-time expansion are given by the impulse responses of the synthesis filters. Therefore, we will concentrate on the synthesis filter bank.

To highlight the main features of the octave-band filter bank expansions, we start with a simple example where we consider a three-stage octave-band filter bank, i.e., $J = 3$, and the filters G_0 and G_1 are Haar filters. The transfer functions of the Haar filters are given as

$$\begin{aligned} G_0(z) &= \frac{1}{\sqrt{2}}(1 + z^{-1}) \\ G_1(z) &= \frac{1}{\sqrt{2}}(1 - z^{-1}) \end{aligned}$$

Using the multirate identity which says that $G(z)$ followed by upsampling by 2 is equivalent to upsampling by 2 followed by $G(z^2)$, we can transform this filter bank into a four-channel filter bank as shown in Figure 3.3. The transfer functions of the equivalent filters are as follows

$$\begin{aligned} G_1^{(1)}(z) &= G_1(z) = \frac{1}{\sqrt{2}}(1 - z^{-1}) \\ G_1^{(2)}(z) &= G_0(z)G_1(z^2) = \frac{1}{2}(1 + z^{-1} - z^{-2} - z^{-3}) \\ G_1^{(3)}(z) &= G_0(z)G_0(z^2)G_1(z^4) \end{aligned}$$

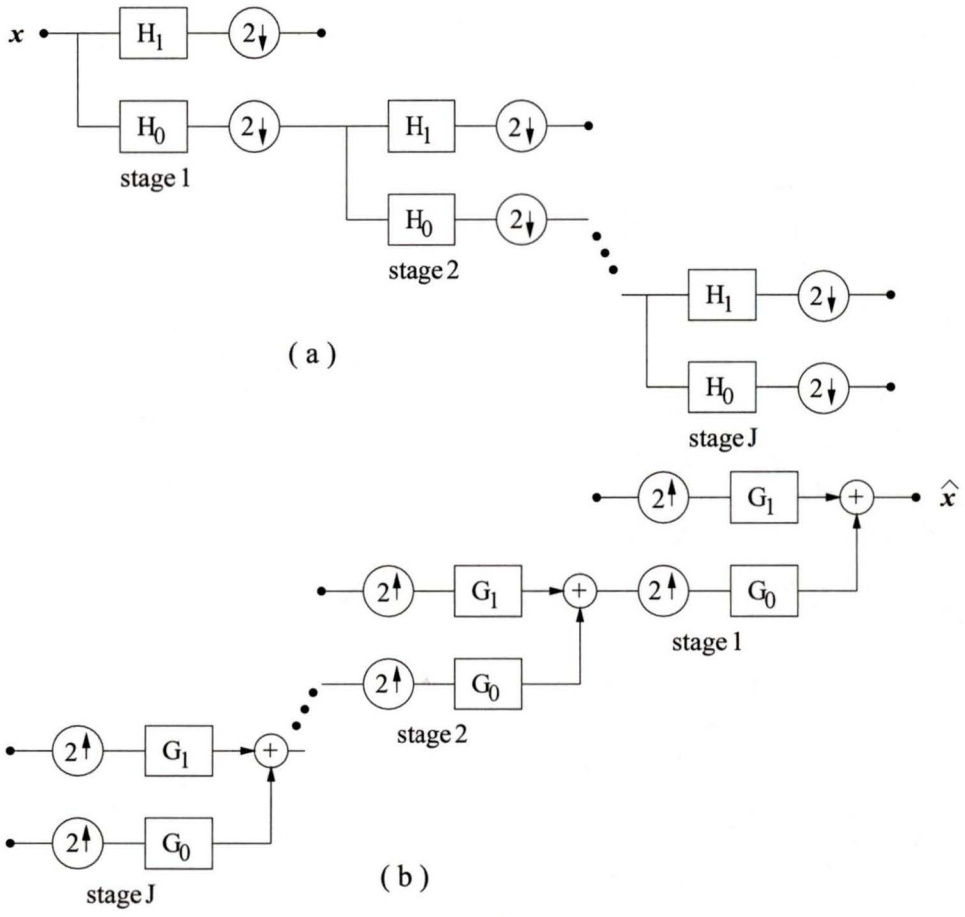


Figure 3.2. *An octave-band filter bank with J stages. (a) Analysis part. (b) Synthesis part.*

$$
\begin{aligned}
&= \frac{1}{2\sqrt{2}}(1 + z^{-1} + z^{-2} + z^3 - z^{-4} - z^{-5} - z^{-6} - z^{-7}) \\
G_0^{(3)}(z) &= G_0(z)G_0(z^2)G_0(z^4) \\
&= \frac{1}{2\sqrt{2}}(1 + z^{-1} + z^{-2} + z^3 + z^{-4} + z^{-5} + z^{-6} + z^{-7})
\end{aligned}
$$

The filters are preceded by upsampling by 2, 4, 8 and 8, respectively.

The equivalent filter comprising three cascaded lowpass filters G_0 each preceded by upsampling of 2, denoted by $G_0^{(3)}$, can be represented by the transfer function

$$G_0^{(3)}(z) = G_0(z^{2^2})G_0^{(2)}(z) = \prod_{k=0}^{2} G_0(z^{2^k})$$

On the other hand, the equivalent filter corresponding to highpass filtering followed

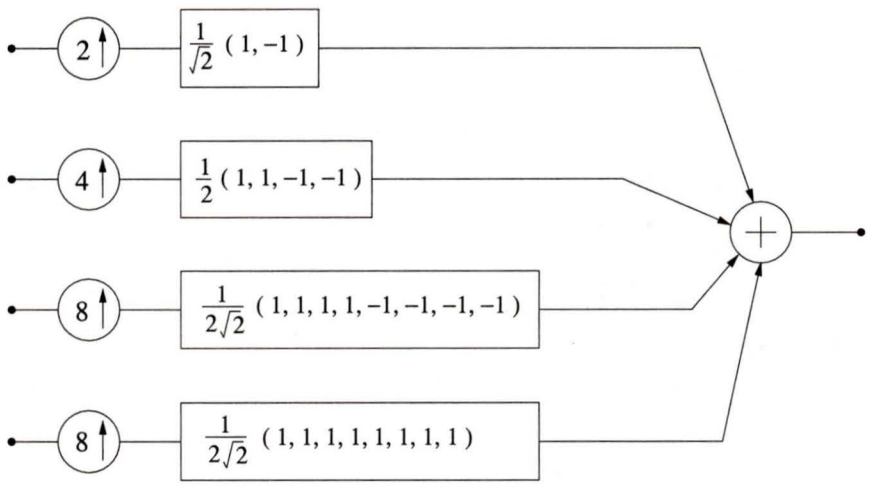


Figure 3.3. *Three-stage octave-band synthesis filter bank with $J = 3$ stages and Haar filters.*

by $(i-1)$ stages of lowpass filtering each preceded by upsampling of 2, denoted by $G_1^{(i)}$ for $i = 1,\ 2,\ 3$ can be represented by

$$G_1^{(i)}(z) \;=\; G_1(z^{2^{i-1}}) \prod_{k=0}^{i-2} G_0(z^{2^k})$$

Since this is an orthonormal system, the time-domain matrices representing analysis and synthesis are just the transposes of each other. Thus the analysis matrix $\mathbf{W}$ representing the actions of the filters $H_1^{(1)},\ H_1^{(2)},\ \ldots,\ H_1^{(J)},\ H_0^{(J)}$ having impulse responses $h_1^{(1)}(n),\ h_1^{(2)}(n),\ \ldots,\ h_1^{(J)}(n),\ h_0^{(J)}(n)$ contains as lines the impulse responses $g_1^{(1)}(n),\ g_1^{(2)}(n),\ \ldots,\ g_1^{(J)}(n),\ g_0^{(J)}(n)$ or $h_1^{(1)}(-n),\ h_1^{(2)}(-n),\ \ldots,\ h_1^{(J)}(-n),\ h_0^{(J)}(-n)$, since analysis and synthesis filters are linked by time reversal. The matrix $\mathbf{W}$ is block-diagonal, i.e.,

$$\mathbf{W} \;=\; \begin{pmatrix} \ddots & & & \\ & \mathbf{A_0} & & \\ & & \mathbf{A_0} & \\ & & & \ddots \end{pmatrix} \tag{3.5}$$

where $J = 3$, G_0 and G_1 are Haar filters and the block $\mathbf{A_0}$ is of the following form:

$$\mathbf{A_0} = \frac{1}{2\sqrt{2}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{pmatrix} \tag{3.6}$$

It can be noted that the matrix reflects the fact that the filter $G_1^{(1)}$ is preceded by upsampling by 2 (the row (2 -2) is shifted by 2 each time and appears 4 times in the matrix). $G_1^{(2)}$ is preceded by upsampling by 4 (the corresponding row is shifted by 4 and appears twice), while filters in $G_1^{(3)}$, $G_0^{(3)}$ are preceded by upsampling by 8 (the corresponding rows appear only once in the matrix) [32].

Next, we consider the case where the filters G_0 and G_1 are Daubechies-4 filters having the transfer functions

$$\begin{aligned} G_0(z) &= 0.4830 + 0.8365z^{-1} + 0.2241z^{-2} - 0.1294z^{-3} \\ G_1(z) &= -0.1294 - 0.2241z^{-1} + 0.8365z^{-2} - 0.4830z^{-3} \end{aligned}$$

For a three-stage two-channel filter bank, the corresponding block $\mathbf{A_0}$ is given as

$$\mathbf{A_0} = [\, \mathbf{R_1} \;\; \mathbf{R_2} \;\; \mathbf{R_3} \;\; \mathbf{R_4} \;\; \mathbf{R_5} \;\; \mathbf{R_6} \;\; \mathbf{R_7} \;\; \mathbf{R_8} \,]^T \tag{3.7}$$

where

$$\begin{aligned} \mathbf{R_1} &\equiv \left[\, h_0^{(3)}(21) \;\; \cdots \;\; h_0^{(3)}(0) \,\right] \\ \mathbf{R_2} &\equiv \left[\, h_1^{(3)}(21) \;\; \cdots \;\; h_1^{(3)}(0) \,\right] \\ \mathbf{R_3} &\equiv \left[\, h_1^{(2)}(9) \;\; \cdots \;\; h_1^{(2)}(0) \;\; 0 \;\; \cdots \;\; 0 \,\right] \\ \mathbf{R_4} &\equiv \left[\, 0 \;\; 0 \;\; 0 \;\; 0 \;\; h_1^{(2)}(9) \;\; \cdots \;\; h_1^{(2)}(0) \;\; 0 \;\; \cdots \;\; 0 \,\right] \\ \mathbf{R_5} &\equiv \left[\, h_1^{(1)}(3) \;\; \cdots \;\; h_1^{(1)}(0) \;\; 0 \;\; \cdots \;\; 0 \,\right] \\ \mathbf{R_6} &\equiv \left[\, 0 \;\; 0 \;\; h_1^{(1)}(3) \;\; \cdots \;\; h_1^{(1)}(0) \;\; 0 \;\; \cdots \;\; 0 \,\right] \\ \mathbf{R_7} &\equiv \left[\, 0 \;\; 0 \;\; 0 \;\; 0 \;\; h_1^{(1)}(3) \;\; \cdots \;\; h_1^{(1)}(0) \;\; 0 \;\; \cdots \;\; 0 \,\right] \end{aligned}$$

$$\mathbf{R_8} \equiv \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & h_1^{(1)}(3) & \cdots & h_1^{(1)}(0) & 0 & \cdots & 0 \end{bmatrix}$$

For a data set of eight samples, the orthonormal discrete wavelet transformation matrix $\mathbf{W}$ which takes the data vector to its wavelet coefficient vector can be constructed by periodization of the data. In other words, block $\mathbf{A_0}$ can be reconstructed into a 8×8 matrix by wrapping over as shown below

$$\mathbf{W} = \mathbf{A_0} = [\, \mathbf{R_1} \;\; \mathbf{R_2} \;\; \mathbf{R_3} \;\; \mathbf{R_4} \;\; \mathbf{R_5} \;\; \mathbf{R_6} \;\; \mathbf{R_7} \;\; \mathbf{R_8} \,]^T \tag{3.8}$$

where

$$\begin{aligned}
\mathbf{R_1} &\equiv \begin{bmatrix} h_{0\,21}^{(3)} + h_{0\,13}^{(3)} + h_{0\,5}^{(3)} & \cdots & h_{0\,16}^{(3)} + h_{0\,8}^{(3)} + h_{0\,0}^{(3)} & h_{0\,15}^{(3)} + h_{0\,7}^{(3)} & h_{0\,14}^{(3)} + h_{0\,6}^{(3)} \end{bmatrix} \\
\mathbf{R_2} &\equiv \begin{bmatrix} h_{1\,21}^{(3)} + h_{1\,13}^{(3)} + h_{1\,5}^{(3)} & \cdots & h_{1\,16}^{(3)} + h_{1\,8}^{(3)} + h_{1\,0}^{(3)} & h_{1\,15}^{(3)} + h_{1\,7}^{(3)} & h_{1\,14}^{(3)} + h_{1\,6}^{(3)} \end{bmatrix} \\
\mathbf{R_3} &\equiv \begin{bmatrix} h_{1\,9}^{(2)} + h_{1\,1}^{(2)} & h_{1\,8}^{(2)} + h_{1\,0}^{(2)} & h_{1\,7}^{(2)} & h_{1\,6}^{(2)} & h_{1\,5}^{(2)} & h_{1\,4}^{(2)} & h_{1\,3}^{(2)} & h_{1\,2}^{(2)} \end{bmatrix} \\
\mathbf{R_4} &\equiv \begin{bmatrix} h_{1\,5}^{(2)} & h_{1\,4}^{(2)} & h_{1\,3}^{(2)} & h_{1\,2}^{(2)} & h_{1\,9}^{(2)} + h_{1\,1}^{(2)} & h_{1\,8}^{(2)} + h_{1\,0}^{(2)} & h_{1\,7}^{(2)} & h_{1\,6}^{(2)} \end{bmatrix} \\
\mathbf{R_5} &\equiv \begin{bmatrix} h_{1\,3}^{(1)} & h_{1\,2}^{(1)} & h_{1\,1}^{(1)} & h_{1\,0}^{(1)} & 0 & 0 & 0 & 0 \end{bmatrix} \\
\mathbf{R_6} &\equiv \begin{bmatrix} 0 & 0 & h_{1\,3}^{(1)} & h_{1\,2}^{(1)} & h_{1\,1}^{(1)} & h_{1\,0}^{(1)} & 0 & 0 \end{bmatrix} \\
\mathbf{R_7} &\equiv \begin{bmatrix} 0 & 0 & 0 & 0 & h_{1\,3}^{(1)} & h_{1\,2}^{(1)} & h_{1\,1}^{(1)} & h_{1\,0}^{(1)} \end{bmatrix} \\
\mathbf{R_8} &\equiv \begin{bmatrix} h_{1\,1}^{(1)} & h_{1\,0}^{(1)} & 0 & 0 & 0 & 0 & h_{1\,3}^{(1)} & h_{1\,2}^{(1)} \end{bmatrix}
\end{aligned}$$

Here $h_{0\,j}^{(3)}$ denotes $h_0^{(3)}(j)$ and $h_{1\,j}^{(i)}$ denotes $h_1^{(i)}(j)$.

For a general case, when the orthogonal octave-band filter bank has J stages, the transfer functions of the equivalent filters are given by

$$G_0^{(J)}(z) = G_0^{(J-1)}(z)\; G_0(z^{2^{J-1}}) = \prod_{K=0}^{J-1} G_0(z^{2^K}) \tag{3.9}$$

$$G_1^{(j)}(z) = G_0^{(j-1)}(z)\; G_1(z^{2^{j-1}}) = G_1(z^{2^{j-1}}) \prod_{K=0}^{j-2} G_0(z^{2^K}) \tag{3.10}$$

$$j = 1,\ 2,\ \ldots,\ J$$

Thus, a data vector with 2^J data samples can be completely decomposed into its wavelet coefficients using a J-stage orthogonal octave-band filter bank. The wavelet transformation matrix $\mathbf{W}$ would then be of size $2^J \times 2^J$.

3.7 Interpolation Results and Discussion

To illustrate the behavior of the interpolation methods presented, we consider a set of irregularly spaced data samples as shown in Figure 3.4(a). The data samples are then approximated to be located at binary rationals $x_{k'} = k/2^6$, $1 \leq k' \leq 20$ for arbitrary integers k, i.e., the data is considered to be of 64 samples, with missing data samples in between.

The data is interpolated using the iterative nonuniform interpolation method for different values of D. After all the missing data points are filled-in in the first stage, we have a data set with samples placed at regular intervals. In the second stage, the DD uniform interpolation process is applied for 6 iterations and the data points are joined to get the interpolated curve. The results for different values of D are compared in Figure 3.5.

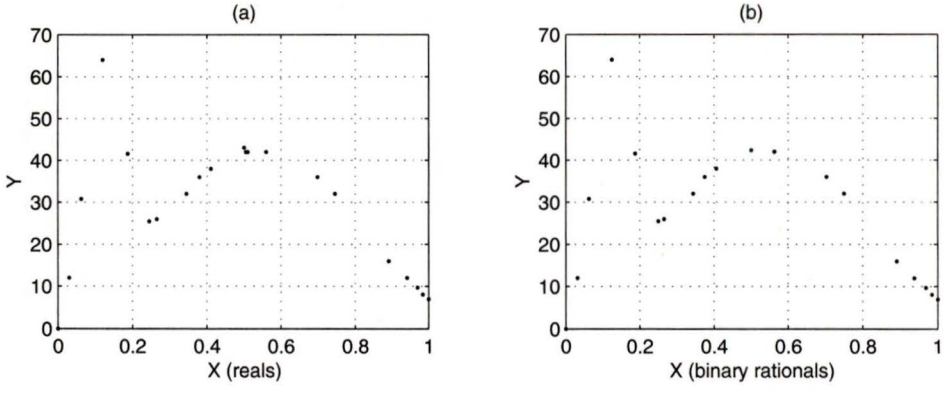


Figure 3.4. *(a) Original data samples. (b) Data samples converted to be placed at binary rationals.*

It can be noted that for $D = 1$, the interpolation is linear as can be observed from Figure 3.5(a) which shows linear characteristics. The curves for $D = 3$ and higher are smoother.

The same set of data samples are then applied to the wavelet-based interpolation process for different sets of filters G_0 and G_1 corresponding to different wavelets. Interpolation results are compared for Daubechies-2 ($D2$) commonly known as Haar filters, Daubechies-4 ($D4$), Daubechies-6 ($D6$) and Daubechies-8 ($D8$) filters in Figure 3.6. While Haar filters are suitable for data with frequent sharp edges and are

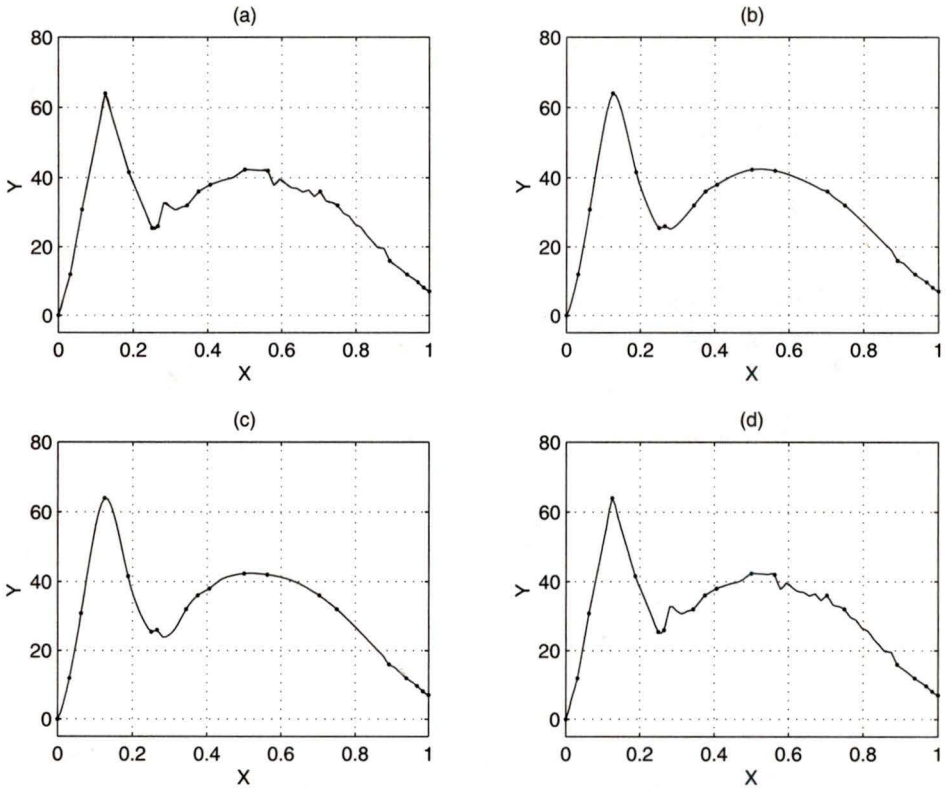


Figure 3.5. *Interpolated curves for* (a) $D = 1$, (b) $D = 3$, (c) $D = 5$ *and* (d) $D = 1$ *for the nonuniform process and* $D = 3$ *for the DD refinement process.*

unsuitable for data which is fairly smooth in nature, $D8$ filters are suitable for data which is fairly smooth in nature. This can be observed from Figure 3.6.

Next, we select a synthetic signal having both smooth and sharp features and randomly choose 96 data samples of the signal as shown in Figure 3.7. The samples are then approximated to be located at $x_{k'} = k/2^8$, for arbitrary integers k, $0 \leq k \leq 256$, such that the final interpolated data has 256 samples. The data is interpolated using just the iterative nonuniform interpolation process with $D = 3$, without using the DD refinement process. The same data is interpolated using the wavelet-based interpolation process. As the final interpolated data has 256 samples, the wavelet transformation matrix $\mathbf{W}$ is of size 256×256. $\mathbf{W}$ is obtained considering an 8-stage orthogonal octave-band filter for a complete wavelet decomposition. The results are shown in Figure 3.8. The root mean square errors between the interpolated data

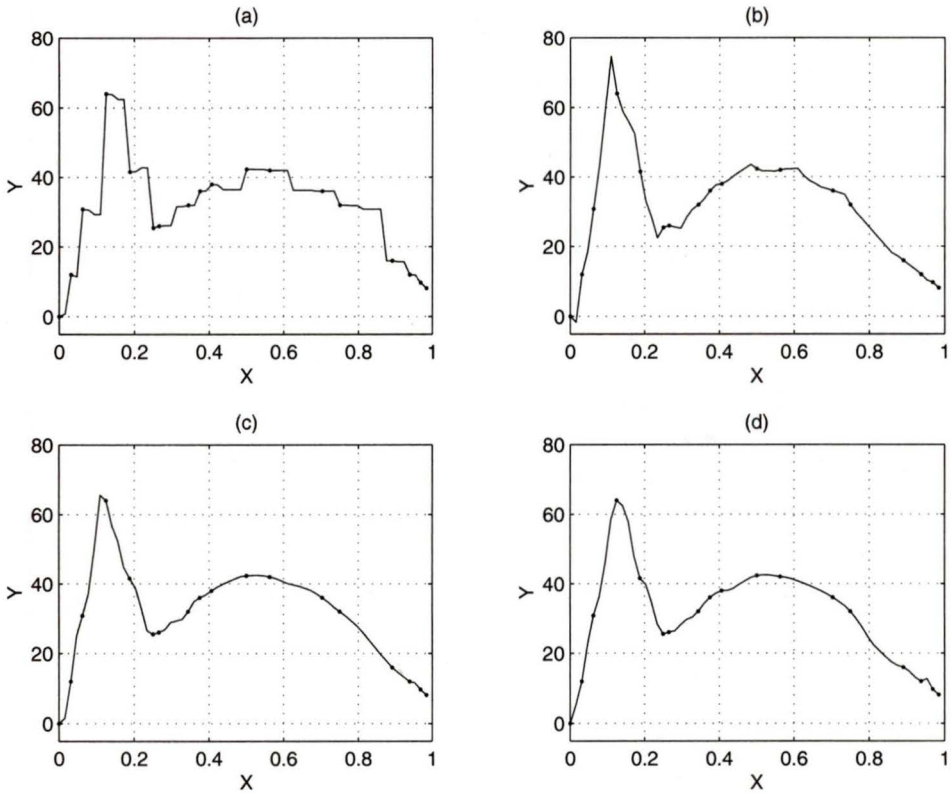


Figure 3.6. *Interpolated curves using (a) Haar, (b) D4, (c) D6 and (d) D8 filters.*

and the original data for the iterative method and the wavelet based method were found to be 0.0273 and 0.0241 respectively. The number of floating point operations required for interpolating the sampled data was found out to be 3997110 whereas the wavelet based method requires a total of 26731415 floating point operations for interpolating the same data.

Similarly, data are interpolated for 512 and 1024 samples as shown in Figures 3.9 and 3.10 respectively. We observe that as the number of interpolated samples increases, i.e., the original sampled data is more sparse, the iterative nonuniform interpolation process gives more inaccurate interpolation results. This happens due to the unavailability of valid data points at higher scales during filling-up of the missing points which forces the process to be executed at coarser scales. At coarser scales the interpolation is not local and thus interpolated samples may be quite inaccurate. As the interpolation is iterative and the data samples used to interpolate missing

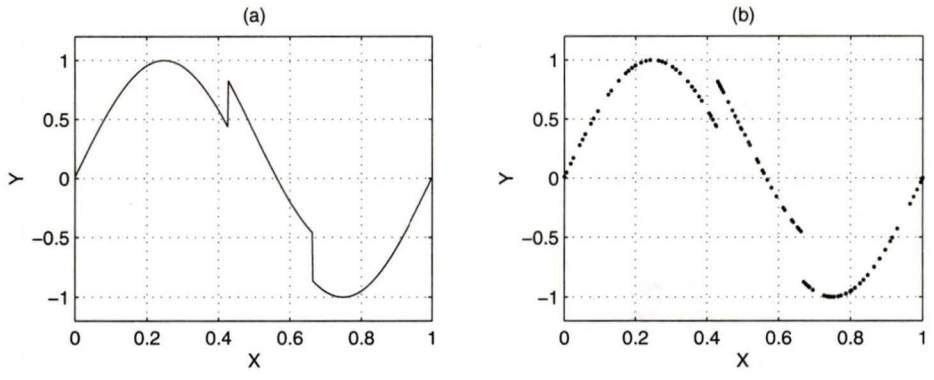


Figure 3.7. *(a) Original signal. (b) Randomly sampled* 96 *data points.*

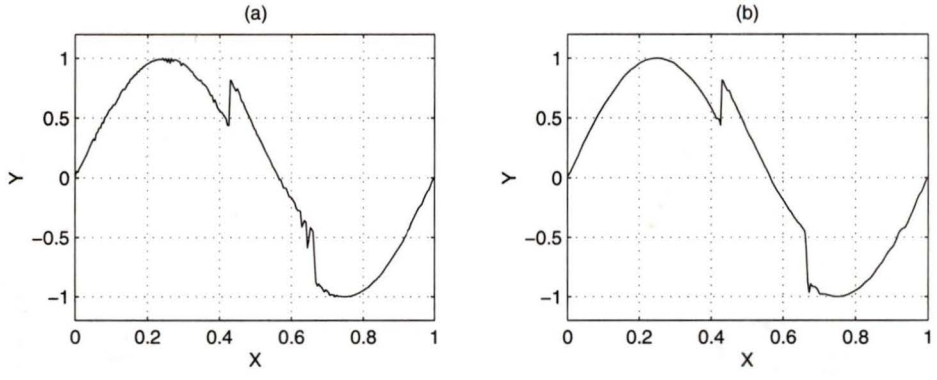


Figure 3.8. *Interpolated data for* 256 *samples using (a) the iterative nonuniform process with* $D = 3$ *and (b) the wavelet-based process.*

samples may be themselves interpolated at coarser scales the error is carried forward rendering the interpolated data highly inaccurate.

For $D = 1$, the interpolation process is simpler as only two data points are needed to interpolate a missing data point instead of four for the case when $D = 3$. As a result more missing data points are filled in at higher scales compared to the case when $D = 3$. However, for $D = 1$ the interpolation is linear and so the interpolated data may not be fairly smooth especially for sparse data sets, as observed earlier in Figure 3.5(a). On the other hand, interpolating the same set of sample points using the wavelet-based interpolation process gives fairly good results in the sense that it preserves, to some extent, the signal characteristics even with a sparse data

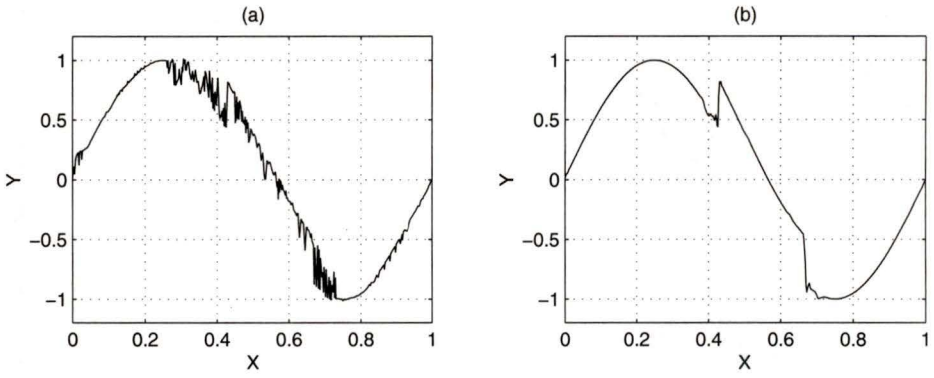


Figure 3.9. *Interpolated data for* 512 *samples using* (*a*) *the iterative nonuniform process with* $D = 3$ *and* (*b*) *the wavelet-based process.*

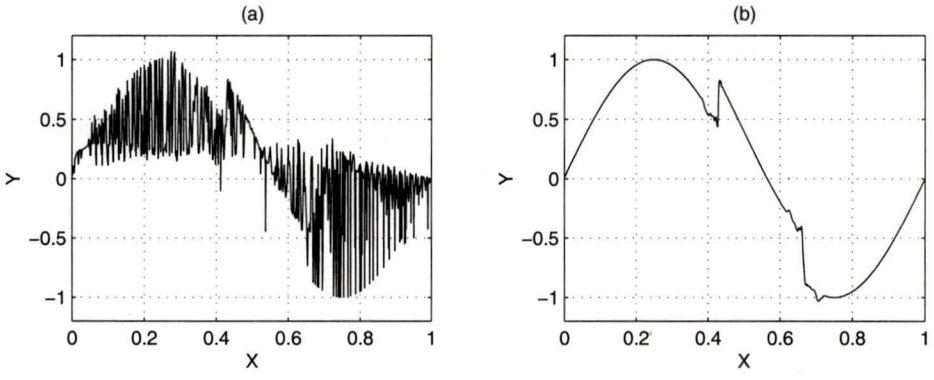


Figure 3.10. *Interpolated data for* 1024 *samples using* (*a*) *the iterative nonuniform process with* $D = 3$ *and* (*b*) *the wavelet-based process.*

set. The iterative nonuniform interpolation process gives inaccurate interpolation at regions with sharp features but the wavelet-based interpolation process maintains even the sharp features of the original data set. This can be observed in Figure 3.9 and Figure 3.10. However, the iterative nonuniform interpolation process and the DD refinement process can be used in conjunction to get better interpolated results at the expense of greater approximation error. For the same data set we obtain 128 samples using the iterative nonuniform interpolation process with $D = 3$ and then apply the DD refinement technique for 3 iterations to obtain an interpolated set of 1024 samples. For comparison, we obtain 256 samples using the iterative nonuniform

interpolation process with $D = 3$ and then apply the DD refinement technique for 2 iterations to obtain an interpolated set of 1024 samples. We compare the results in Figure 3.11. It can be noted that the smaller the number of data samples in the estimated output signal the greater is the approximation error.

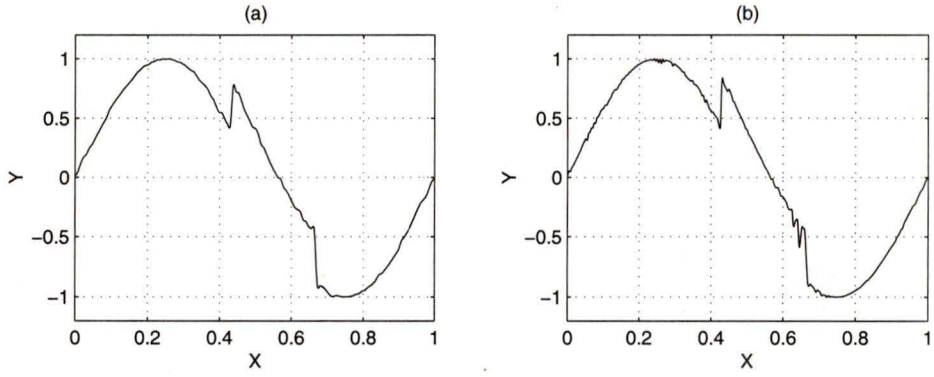


Figure 3.11. *Interpolated data obtained for* 1024 *samples* (*a*) *using the iterative nonuniform process with* $D = 3$ *for* 128 *samples and then applying the DD refinement technique for* 3 *iterations,* (*b*) *using the iterative nonuniform process with* $D = 3$ *for* 256 *samples and then applying the DD refinement technique for* 2 *iterations.*

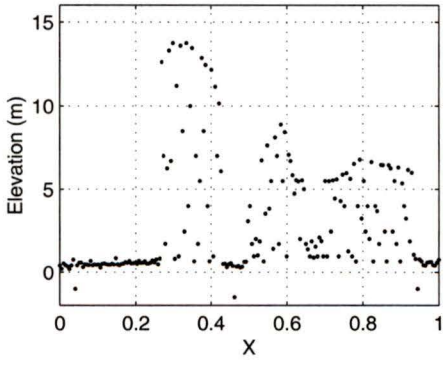


Figure 3.12. *LIDAR profiler data.*

We next implement the interpolation processes for a LIDAR profiler data set shown in Figure 3.12. The profiler data points are initially categorized as ground hits or vegetation-top hits (Figure 3.13). The categorized data points are then interpolated

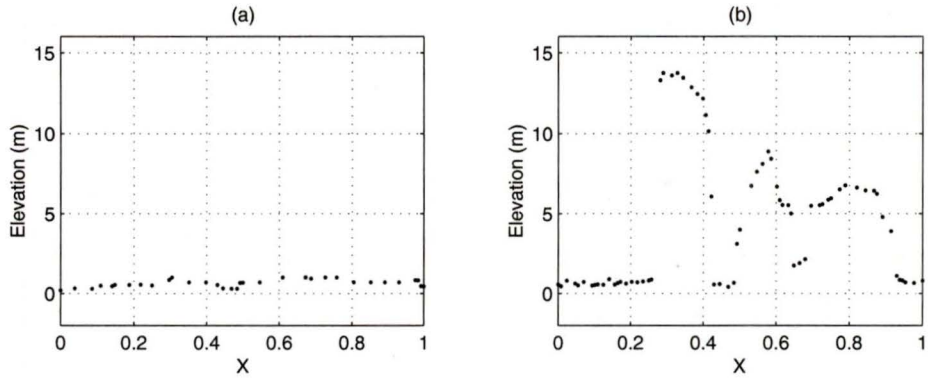


Figure 3.13. *Data points categorized as (a) ground hits and (b) vegetation top hits.*

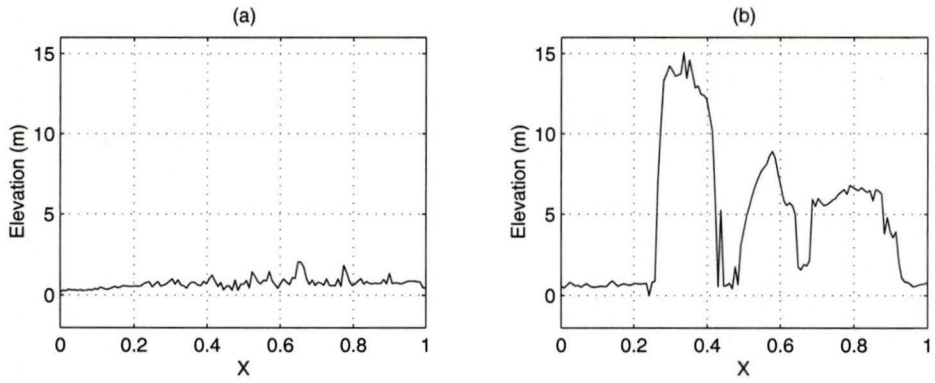


Figure 3.14. *Interpolated data using the iterative nonuniform interpolation process for (a) ground profile and (b) vegetation top profile.*

to get the ground and vegetation-top profiles using the iterative interpolation process and the wavelet-based process as shown in Figure 3.14 and Figure 3.15, respectively.

Although the wavelet-based interpolation process produces better results compared to the iterative scheme in the sense that it produces smoother interpolated data, the wavelet approach has some limitations. For an interpolated data of M samples, the wavelet transformation matrix $\mathbf{W}$ is of size $M \times M$. For a large M, the computational time to formulate $\mathbf{W}$ and also for the solution given by (3.4) is very high. Moreover, the smaller the number of missing data points, the more is the computational time for the solution given by (3.4). On the other hand, for the iterative scheme the smaller the number of missing data points, the faster is the interpolation

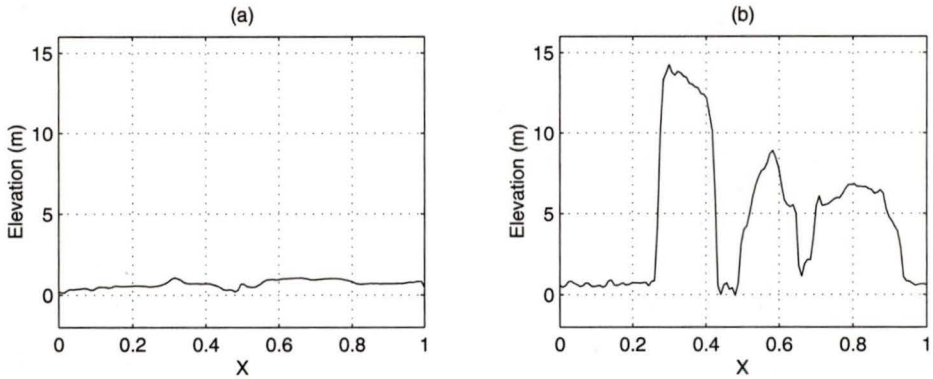


Figure 3.15. *Interpolated data using the wavelet-based nonuniform interpolation process for (a) ground profile and (b) vegetation top profile.*

process.

3.8 Conclusions

Two techniques for interpolating nonuniformly sampled data points have been presented. The first method interpolates the data using an iterative scheme whereas the second method is based on wavelet domain processing. As both the methods are multiscale interpolation processes the original data samples are initially approximated to be located at binary rationals.

The iterative nonuniform interpolation process has been considered as a two stage process. At the first stage, the data is regularized by filling-in the missing data points using an iterative scheme. When all the missing data points are filled-in the problem simplifies to that of a uniform interpolation problem. The uniform interpolation process iteratively interpolates data at halfway between the uniformly spaced data points.

The second interpolation process estimates the original signal from the given signal samples by minimizing the Sobolev norm of the interpolated signal. This reduces to a weighted least-square optimization problem, in the wavelet domain.

Interpolation results for both the methods have been compared and their behavior studied. Finally the interpolation methods have been used to obtain the ground and

vegetation-top profiles for a LIDAR profiler data.

It was found that the Sobolev-norm approach yields better results but at a significantly higher computational cost.

Chapter 4

Signal Processing for a LIDAR Scanner Data

4.1 Introduction

LIDAR scanner data can be considered as 2-D data with the projections of the data points scattered on the x-y plane. In order to get the ground and canopy profiles, data points have to be interpolated at a regular rectangular grid. We, therefore, have a 2-D nonuniform interpolation problem. Although numerous methods have been proposed for the purpose of 2-D nonuniform interpolation, we will concentrate on two approaches: one based on interpolation over a triangulated network and the other based on interpolation in the wavelet domain.

Triangle-based interpolation has gained extensive use, particularly for topographical mapping. In this approach, a triangulated network is formed with the data points as the vertices of the network. The resulting surface is modeled as a combination of surfaces, each surface modeled over a triangle in the triangulated network. Although there may be many different ways in which the mesh of triangles can be drawn from a set of data points, attempts have been made to construct an unique, *optimal* triangulated network. This implies that the triangles thus formed should be as close to equiangular triangles as possible. Thin triangles are not preferred as they usually give rise to inaccurate interpolation results.

Lawson [33] introduced a procedure to optimize the triangulation network. The resulting triangulated network thus formed is a Delaunay triangulation, wherein the triangles are as equiangular as possible [34]-[35]. One of the advantages of Delaunay triangulation, as opposed to triangulations constructed heuristically, is that they

automatically avoid forming triangles with small included angles whenever this is possible. Lawson [33] and Sibson [35] have shown that Delaunay triangulations are, by definition, locally equiangular and the triangles in the triangulated network satisfy the *max-min* criterion. This means that for every convex quadrilateral formed by two adjacent triangles, the minimum of the six angles in the two triangles is greater than it would have been if the alternative diagonal had been drawn and the other pair of triangles chosen. For this property, Delaunay triangulations are particularly suited to grid generation for finite element and contouring algorithms. Sloan [36] introduced a fast algorithm to construct Delaunay triangulations on the plane. This procedure starts with one triangle and then adds one data point at a time into the triangulation, maintaining the *max-min* criterion at each time. This procedure will be discussed in detail in sections 4.2-4.3. Subsequent improvements over Sloan's procedure were proposed by Huang and Shih [37].

Once the construction of the triangulated network is complete, a curved surface can be fitted to each triangle cell by using a low-order bivariate polynomial. Zlámal [38] constructed a bivariate 5th-degree polynomial over each triangular cell to approximate the surface over the cell. The procedure ensures C^1 continuity. Moreover, the polynomials have piecewise continuous second-order derivatives. The procedure is discussed in futher detail in sections 4.4-4.6.

A multiscale approach to maximum-smoothness interpolation was proposed by Choi and Baraniuk [27]. This process was discussed in detail for a 1-D signal in sections 3.4-3.6. The technique can be applied to a 2-D signal by converting it into a 1-D signal which contains the subsequent rows of the 2-D signal. However, the wavelet coefficients formed using the wavelet transformation matrix $\mathbf{W}$ should be those of the 2-D signal and not of the 1-D signal which contains the subsequent rows of the 2-D signal. Thus the formulation of the wavelet transformation matrix $\mathbf{W}$ is different for a 2-D signal than its 1-D counterpart. Section 4.7 discusses the formulation of the wavelet transformation matrix which converts a 2-D signal into its corresponding wavelet coefficients.

4.2 The Delaunay Triangulation Process

The algorithm proposed by Sloan [36] combines features of the Watson [39] and the Lawson [33] procedures. The process is started by considering a single *super-triangle* which completely encompasses all of the data points to be triangulated. Initially the Delaunay triangulation comprises the *super-triangle*. The Delaunay triangulation is then assembled by introducing each data point, one at a time, into the existing Delaunay triangulation which is then updated.

When a new point P is introduced into the triangulation the existing triangle which encloses P is found and three new triangles are formed by connecting P to each of its vertices. After the new point P has been inserted, the existing triangulation is updated to a Delaunay triangulation using the swapping algorithm of Lawson [33]. In this procedure all the triangles which are adjacent to the edges opposite P are placed on a last-in first-out stack. Each triangle is then unstacked, one at a time. If P lies inside the circumcircle of the unstacked triangle, then the diagonal of the convex quadrilateral formed by the triangle containing P as a vertex and the adjacent triangle is in the wrong direction according to the *max-min* criterion of a Delaunay triangulation and so it is replaced by the alternative diagonal using a swapping procedure to preserve the structure of the Delaunay triangulation. Once the swap is completed, triangles which are opposite P are added to the stack. The next triangle is then unstacked and the whole process is repeated until the stack is empty. This results in a new Delaunay triangulation containing the point P. Figure 4.1 illustrates the swapping procedure. It can be noted that if P lies outside (or on) the circumcircle for a stacked triangle, then no action is taken and we simply skip to the next triangle on the stack. It has been shown by Lawson [33] that this iterative algorithm must result in a Delaunay triangulation and will always terminate after a finite number of swaps.

After all the data points have been added to the triangulation, the final Delaunay triangulation is obtained by removing all of the triangles that contain one or more of the *super-triangle* vertices. Any vertex which appears in these deleted triangles, but is not a *super-triangle* vertex, must lie on the boundary of the triangulation.

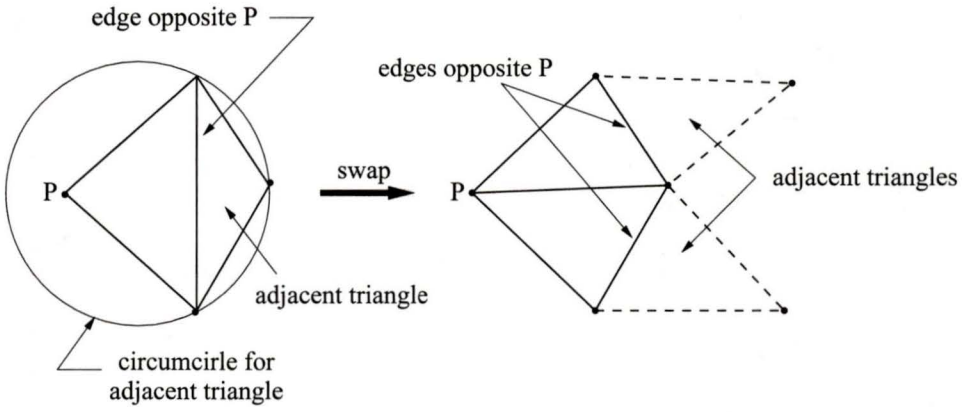


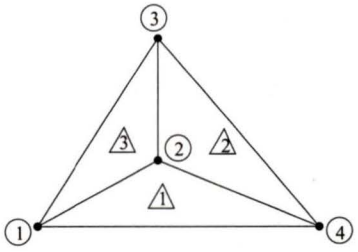
Figure 4.1. *Lawson's swapping algorithm.*

4.3 Implementation of the Procedure

Initially, the x and y coordinates of the data points are stored in real vectors X and Y. Throughout the process the structure of the Delaunay triangulation is stored in the integer arrays V and E. Both of these arrays are two-dimensional with V containing the vertices for each triangle and E containing the indices to the adjacent triangles. The conventions for this data structure are shown for a simple example in Figure 4.2.

To begin the process of triangulation we define the vertices and coordinates for the *super-triangle*. The *super-triangle* is initially stored in the first column of the vertex and adjacency arrays, and its vertex coordinates are stored in the last three locations of the X and Y vectors. It can be noted that if the corners of the *super-triangle* are very close to the window enclosing the points, the boundary of the final triangulation may be locally concave. Strictly speaking, such a triangulation does not correspond exactly to a Delaunay triangulation, since some long thin triangles along the boundary might have been omitted. Therefore, it should be assured that the *super-triangle* is sufficiently larger than the window enclosing the data points.

After the *super-triangle* has been defined, each of the points is inserted into the triangulation one at a time. To insert a point P, a search is done to find an existing triangle T which encloses P. The search is started at the triangle which has been most recently created, and checks if the point is to the right of any of its edges. Since the triangle vertices are always listed in an anticlock-wise sequence, a point can only be enclosed by a triangle if it is to the left of each of its edges. If the point lies to the



triangle	1	2	3
vertices	1	2	3
	4	4	1
	2	3	2

V

(1) vertices listed anticlockwise

(2) first vertex at point of contact of first and third adjacent triangles

triangle	1	2	3
adjacent triangles	0	1	0
	2	0	1
	3	3	2

E

(1) adjacent triangles listed anticlockwise

(2) zero denotes no adjacent triangle

Figure 4.2. *Data structure for triangulation algorithm.*

right of any edge of the triangle, then the search shifts to the triangle which is adjacent to this edge and the process is repeated. In this way the search marches from one triangle to the next in the general direction of the point. This searching algorithm is due to Lawson and avoids the need to search all of the triangles in the grid. Figure 4.3 illustrates the searching algorithm. After triangle T has been identified, it is deleted and three new triangles are created by connecting P to each of its vertices. Next, each triangle containing P as a vertex is placed on a last-in first-out stack (provided that the edge opposite P is adjacent to some other triangle). This completes the initial insertion phase, and we are now ready to update the triangulation to a Delaunay triangulation using the swapping algorithm of Lawson [33].

Lawson's procedure is illustrated in Figure 4.4. Triangle L is the triangle removed from the stack. Triangle R is the triangle opposite point P which is adjacent to L. Triangles L and R share the edge $V1 - V2$ and form a quadrilateral with vertices P, $V2$, $V3$, $V1$. If the point P is inside the circumcircle of triangle R, then the diagonal $V1 - V2$ needs to be replaced by the diagonal $P - V3$ to preserve the structure of the Delaunay triangulation. As pointed out by Lawson, this circumcircle

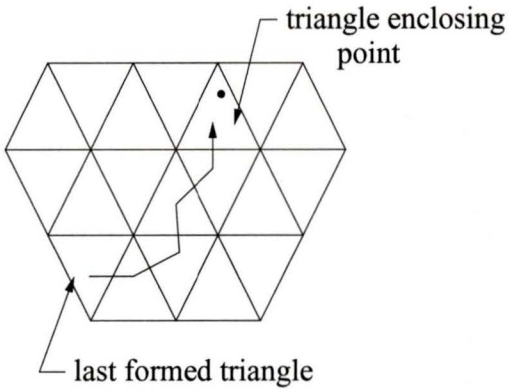


Figure 4.3. *Search path to find the triangle which encloses a data point.*

check maximizes the minimum angle occuring in any pair of adjacent triangles forming a convex quadrilateral. If a swap is necessary, as shown in Figure 4.4, the vertex and adjacent arrays for triangles L and R are updated, as are the adjacency arrays for triangles A and C. Provided that there is a triangle opposite P which is adjacent to L, triangle L is placed on the stack. Similarly for triangle R. The next triangle is then removed from the stack and the whole process is repeated until the stack is empty. This signifies that the insertion of P is complete and the new triangulation is a Delaunay triangulation.

After all of the points have been triangulated, the program deletes any triangle which contains one or more *super-triangle* vetices. During this phase the vertex and adjacency arrays are updated to fill any blanks created. The vertex and adjacent triangle lists are stored in V and E, respectively.

Figure 4.5 shows the Delaunay triangulation of 12 points. It can be observed that each of the quadrilaterals, formed by joining two adjacent triangles of the Delaunay triangulation satisfies the *max-min* criterion, i.e., the least angle among the six angles of the two adjacent triangles is greater than that of the least angle among the six angles which would have been formed with the other diagonal of the quadrilateral.

4.4 Bivariate-Interpolation in a Triangular Cell

Assuming that the plane is divided into a number of triangles, the procedure discussed by Zlámal [38] interpolates the values of a function in each triangle in the triangulated

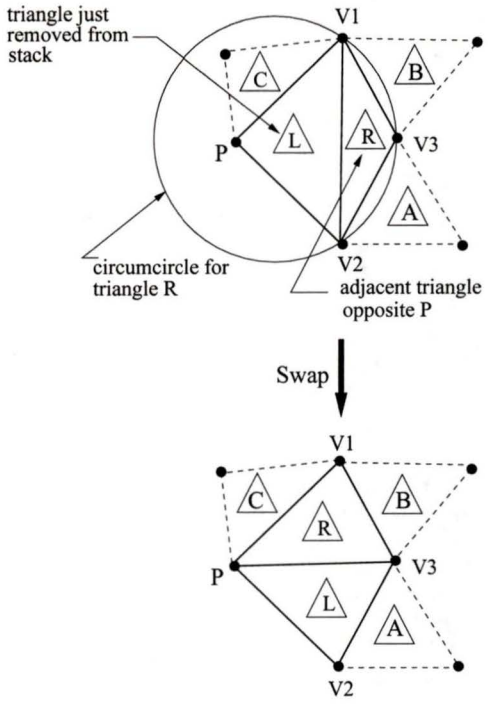


Figure 4.4. *Implementation of Lawson's swapping algorithm.*

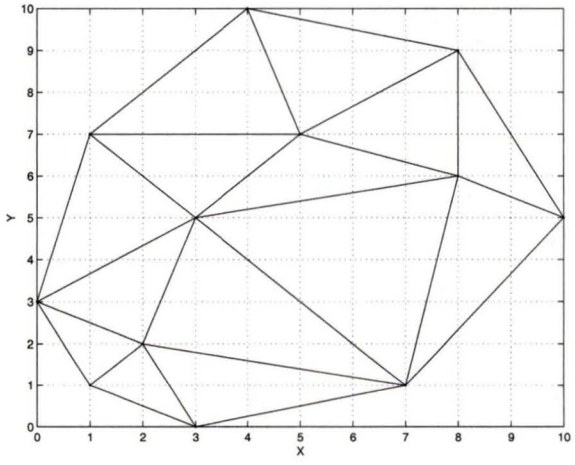


Figure 4.5. *Delaunay triangulation of* 12 *data points.*

network. For each triangle T, let the vertices of T be denoted by P_i, $i = 1, 2, 3$ and the mid-points of the sides of T be denoted as Q_i, $i = 1, 2, 3$. The value of the function at point (x, y) in the triangle T is interpolated by a bivariate fifth-degree

polynomial in x and y given by

$$z(x,y) = \sum_{j=0}^{5} \sum_{k=0}^{5-j} q_{jk}\, x^j\, y^k \tag{4.1}$$

where q_{jk} are the coefficients of the polynomial to be determined using the values of z and $\partial z/\partial x$, $\partial z/\partial y$, $\partial^2 z/\partial^2 x^2$, $\partial^2 z/\partial^2 y^2$ and $\partial^2 z/\partial x \partial y$, denoted as z_x, z_y, z_{xx}, z_{yy} and z_{xy}, respectively, at P_i. This determines 18 coefficients and the remaining three coefficients are determined by considering three normal derivatives $\partial z/\partial v$ at Q_i, where v is the normal to the side of T under consideration. The interpolating function is a smooth function when the interpolating function and its first-order partial derivatives are continuous on the boundaries of the triangles. In order to make the first derivatives continuous on the sides of the triangles, Akima [40] imposed a condition that the partial derivative of the function differentiated in the direction of v is a polynomial of degree three, at most, in the variable measured in the direction of the side of the triangle. In other words, when the coordinate system is transformed to another Cartesian system, the s-t system [40], in such a way that the s axis is parallel to each of the side of the triangle, z_t is a polynomial of degree three at most in s and so it satisfies

$$z_{tssss} = 0 \tag{4.2}$$

Since a triangle has three sides, this condition yields the three remaining coefficients.

The partial derivatives are estimated in two steps; the first-order derivatives z_x and z_y in the first step and the second-order derivatives z_{xx}, z_{yy} and z_{xy} in the second step. To estimate the first-order derivative at point P_0, several additional data points P_i, $i = 1, 2, \ldots, N$ are selected, the projections of which are nearest to the projection of P_0. Two data points P_i and P_j are taken out of the N points and the vector product of P_0P_i and P_0P_j is constructed in such a way that the resulting vector product points upwards, i.e., the z component of the vector product is positive. Vector products for all possible combinations of P_0P_i and P_0P_j $(i \neq j)$ are constructed and the vector sum of all the vector products thus constructed is taken. The first-order partial derivatives z_x and z_y are then estimated as those of a plane that is normal to the resultant vector sum thus composed.

In the second step, the same procedure as described above is applied to the estimated z_x values at P_i, $i = 1, 2, \ldots, N$ and the values of $z_{xx} = (z_x)_x$ and $z_{xy} = (z_x)_y$

are estimated at P_0. The procedure is repeated for the estimated z_y values to estimate the values of $z_{yx} = (z_y)_x$ and $z_{yy} = (z_y)_y$ at P_0. The second derivative z_{xy} at P_0 is then estimated as the average of z_{yx} and z_{xy} computed earlier.

4.5 Formulation of the Polynomial

We denote the vertices of the triangle by P_1, P_2 and P_3 in a counter-clockwise order, and their respective coordinates in the x-y Cartesian coordinate system by (x_1, y_1), (x_2, y_2) and (x_3, y_3) as shown in Figure 4.6(a). To facilitate the determination of the coefficients of the triangle a new coordinate system, the u-v system is defined. The vertices P_1, P_2 and P_3 are represented by $(0, 0)$, $(1, 0)$ and $(0, 1)$, respectively, as shown in Figure 4.6(b). From the relationship between the x-y system and the u-v system, the five partial derivatives z_u, z_v, z_{uu}, z_{vv} and z_{uv} at each vertex of the triangle can be obtained. The fifth-degree polynomial in the x-y system is then transformed into the u-v system.

The coordinate transformation between the x-y system and the u-v system is represented by

$$\begin{aligned} x &= au + bv + x_1 \\ y &= cu + dv + y_1 \end{aligned} \tag{4.3}$$

where

$$\begin{aligned} a &= x_2 - x_1 \\ b &= x_3 - x_1 \\ c &= y_2 - y_1 \\ d &= y_3 - y_1 \end{aligned} \tag{4.4}$$

The inverse relation is given by

$$\begin{aligned} u &= \frac{d(x - x_1) - b(y - y_1)}{(ad - bc)} \\ v &= \frac{-c(x - x_1) + a(y - y_1)}{(ad - bc)} \end{aligned} \tag{4.5}$$

Using (4.3), the partial derivatives in the x-y system can be transformed to those in the u-v system. The first derivative z_u can be expressed as

$$\frac{\partial z}{\partial u} = \frac{\partial z}{\partial x}\frac{\partial x}{\partial u} + \frac{\partial z}{\partial y}\frac{\partial y}{\partial u}$$

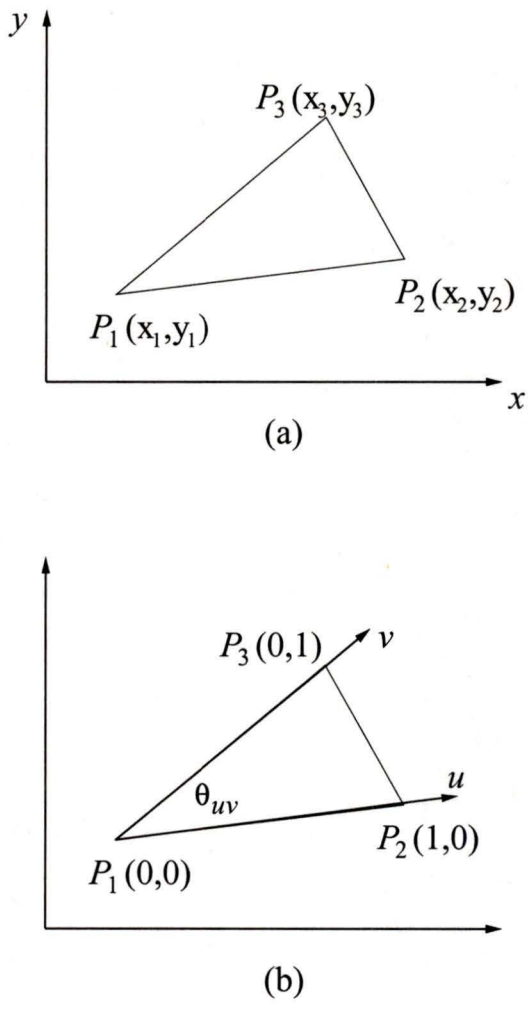


Figure 4.6. (*a*) *The x-y Cartesian coordinate system.* (*b*) *The u-v coordinate system.*

i.e.,

$$z_u = a\, z_x + c\, z_y$$

Similarly, the other first-order and second-order derivatives are obtained as

$$\begin{aligned} z_v &= b\, z_x + d z_y \\ z_{uu} &= a^2\, z_{xx} + 2ac\, z_{xy} + c^2\, z_{yy} \\ z_{uu} &= ab\, z_{xx} + (ad + bc)\, z_{xy} + cd\, z_{yy} \\ z_{vv} &= b^2\, z_{xx} + 2bd\, z_{xy} + d^2\, z_{yy} \end{aligned} \tag{4.6}$$

Since the coordinate transformation from the x-y system to the u-v system is linear, the interpolating polynomial given by (4.1) is transformed to

$$z(u,v) = \sum_{j=0}^{5} \sum_{k=0}^{5-j} p_{jk}\, u^j\, v^k \tag{4.7}$$

Since coefficients p_{jk} are determined directly and are used for interpolating the z values, it is unnecessary to relate coefficients p_{jk} to coefficients q_{jk} used in (4.1).

The partial derivatives of $z(u,v)$ in the u-v system can be expressed as

$$\begin{aligned}
z_u(u,v) &= \sum_{j=1}^{5} \sum_{k=0}^{5-j} j\, p_{jk}\, u^{j-1}\, v^k \\
z_v(u,v) &= \sum_{j=0}^{4} \sum_{k=1}^{5-j} k\, p_{jk}\, u^j\, v^{k-1} \\
z_{uu}(u,v) &= \sum_{j=2}^{5} \sum_{k=0}^{5-j} j(j-1)\, p_{jk}\, u^{j-2}\, v^k \\
z_{uv}(u,v) &= \sum_{j=1}^{4} \sum_{k=1}^{5-j} jk\, p_{jk}\, u^{j-1}\, v^{k-1} \\
z_{vv}(u,v) &= \sum_{j=0}^{3} \sum_{k=2}^{5-j} k(k-1)\, p_{jk}\, u^j\, v^{k-2}
\end{aligned} \tag{4.8}$$

We denote the lengths of the unit vectors in the u-v system (i.e., the lengths of sides P_1P_2 and P_1P_3 by L_u and L_v, respectively, and the angle between the u and v axes by θ_{uv}. They are given by

$$\begin{aligned}
L_u &= \sqrt{a^2 + c^2} \\
L_v &= \sqrt{b^2 + d^2} \\
\theta_{uv} &= \tan^{-1}\left(\frac{d}{b}\right) - \tan^{-1}\left(\frac{c}{a}\right)
\end{aligned} \tag{4.9}$$

where a, b, c and d are constants given in (4.3).

To utilize the smoothness condition given by (4.2), we rotate the s-t system. First, we consider the case where the s axis is parallel to side P_1P_2, as shown in Figure 4.7. The coordinate transformation between the u-v system and the s-t system can be expressed by

$$u = \frac{(\sin\theta_{uv})(s - s_0) - (cos\theta_{uv})(t - t_0)}{L_u \sin\theta_{uv}}$$

$$v = \frac{t - t_0}{L_v \sin\theta_{uv}} \tag{4.10}$$

where (s_0, t_0) is the vertex position of P_1, L_u, L_v and θ_{uv} are as given in (4.9).

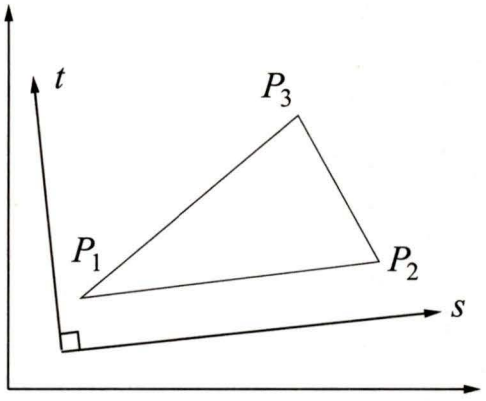


Figure 4.7. *The s-t coordinate system with the s axis parallel to P_1P_2.*

Partial differentiation of $z = z(u, v)$ in (4.7) with respect to t gives

$$\begin{aligned} \frac{\partial z}{\partial t} &= \frac{\partial z}{\partial u}\frac{\partial u}{\partial t} + \frac{\partial z}{\partial v}\frac{\partial v}{\partial t} \\ &= -\frac{\cos\theta_{uv}}{L_u \sin\theta_{uv}} z_u + \frac{1}{L_v \sin\theta_{uv}} z_v \end{aligned} \tag{4.11}$$

The partial derivative of $z = z(u, v)$ with respect to s using (4.10) is given by

$$\begin{aligned} \frac{\partial z}{\partial s} &= \frac{\partial z}{\partial u}\frac{\partial u}{\partial s} + \frac{\partial z}{\partial v}\frac{\partial v}{\partial s} \\ &= \frac{1}{L_u}\frac{\partial z}{\partial u} \end{aligned} \tag{4.12}$$

Partially differentiating (4.11) four times with respect to s using (4.12) we obtain

$$z_{tssss} = \frac{1}{L_u^4}\left(\frac{-\cos\theta_{uv}}{L_u \sin\theta_{uv}} z_{uuuuu} + \frac{1}{L_v \sin\theta_{uv}} z_{vuuuu}\right) \tag{4.13}$$

z_{uuuuu} and z_{vuuuu} as obtained by partially differentiating the polynomial z_{uv} in (4.8) with respect to u and v are given as

$$\begin{aligned} z_{uuuuu} &= 120 p_{50} \\ z_{vuuuu} &= 24 p_{41} \end{aligned} \tag{4.14}$$

Substituting (4.14) into (4.13) and letting $z_{tssss} = 0$, we get

$$L_u \; p_{41} - 5L_v \cos\theta_{uv} \; p_{50} = 0 \tag{4.15}$$

Next, we consider the case where the s axis is parallel to side P_1P_3, as shown in Figure 4.8. The coordinate transformation is expressed by

$$\begin{aligned} u &= -\frac{t - t_0}{L_u \; \sin\theta_{uv}} \\ v &= \frac{(\sin\theta_{uv})(s - s_0) + (cos\theta_{uv})(t - t_0)}{L_v \; \sin\theta_{uv}} \end{aligned} \tag{4.16}$$

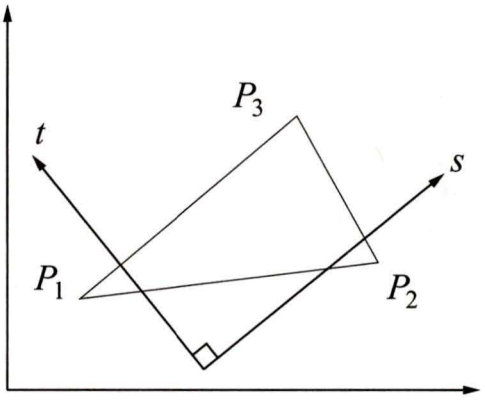


Figure 4.8. *The s-t coordinate system with the s axis parallel to* P_1P_3.

In this case

$$z_{tssss} = \frac{1}{L_v^4}\left(\frac{-1}{L_u \; \sin\theta_{uv}} z_{uvvvv} + \frac{\cos\theta_{uv}}{L_v \; \sin\theta_{uv}} z_{vvvvv}\right) \tag{4.17}$$

and the derivatives are obtained as

$$\begin{aligned} z_{uvvvv} &= 24p_{14} \\ z_{vvvvv} &= 120p_{05} \end{aligned} \tag{4.18}$$

Substituting (4.18) into (4.17) and letting $z_{tssss} = 0$, we get

$$L_v \; p_{14} - 5L_u \cos\theta_{uv} \; p_{05} = 0 \tag{4.19}$$

Next, we consider the third case where the s axis is parallel to side P_2P_3, as shown in Figure 4.9. The coordinate transformation is expressed by

$$\begin{aligned} u &= A\,(s - s_0) + B\,(t - t_0) \\ v &= C\,(s - s_0) + D\,(t - t_0) \end{aligned} \tag{4.20}$$

where

$$\begin{aligned} A &= \frac{\sin(\theta_{uv} - \theta_{us})}{L_u \sin\theta_{uv}} \\ B &= -\frac{\cos(\theta_{uv} - \theta_{us})}{L_u \sin\theta_{uv}} \\ C &= \frac{\sin\theta_{us}}{L_v \sin\theta_{uv}} \\ D &= \frac{\cos\theta_{us}}{L_v \sin\theta_{uv}} \\ \theta_{us} &= \tan^{-1}\left(\frac{d-c}{b-a}\right) - \tan^{-1}\left(\frac{c}{a}\right) \end{aligned} \tag{4.21}$$

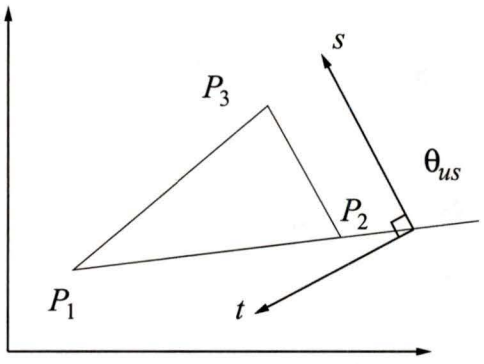


Figure 4.9. *The s-t coordinate system with the s axis parallel to P_2P_3.*

The result for this case is complex and is given by

$$\begin{aligned} &5A^4Bp_{50} + A^3(4BC + AD)p_{41} + A^2C(3BC + 2AD)p_{32} \\ &+AC^2(2BC + 3AD)p_{23} + C^3(BC + 4AD)p_{14} + 5C^4Dp_{05} = 0 \end{aligned} \tag{4.22}$$

Equations (4.15), (4.19) and (4.22) are the results of implementation of the smoothness condition given by (4.2) and are used to determine the coefficients of the polynomial $z = z(u, v)$.

4.6 Determination of the Coefficients of the Polynomial

We can determine the coefficients of the lower-power terms by letting $u = 0$ and $v = 0$ and inserting the values of z, z_u, z_v, z_{uu}, z_{uv} and z_{vv} at P_1 (i.e., $u = 0$ and $v = 0$) in

(4.7) and (4.8). Doing so, we obtain

$$\begin{aligned} p_{00} &= z(0,0) \\ p_{10} &= z_u(0,0) \\ p_{01} &= z_v(0,0) \\ p_{20} &= z_{uu}(0,0)/2 \\ p_{11} &= z_{uv}(0,0) \\ p_{02} &= z_{vv}(0,0)/2 \end{aligned} \tag{4.23}$$

Next, letting $u = 1$ and $v = 0$ and inserting the values of z, z_u and z_{uu} at P_2 (i.e., $u = 1$ and $v = 0$) in (4.7) and the first and the third equations in (4.8), we obtain the following three equations

$$\begin{aligned} p_{30} + p_{40} + p_{50} &= z(1,0) - p_{00} - p_{10} - p_{20} \\ 3p_{30} + 4p_{40} + 5p_{50} &= z_u(1,0) - p_{10} - 2p_{20} \\ 6p_{30} + 12p_{40} + 20p_{50} &= z_{uu}(1,0) - 2p_{20} \end{aligned}$$

Solving these equations with respect to p_{30}, p_{40} and p_{50}, we obtain

$$\begin{aligned} p_{30} &= [20z(1,0) - 8z_u(1,0) + z_{uu}(1,0) - 20p_{00} - 12p_{10} - 6p_{20}]/2 \\ p_{40} &= -15z(1,0) + 7z_u(1,0) - z_{uu}(1,0) + 15p_{00} + 8p_{10} + 3p_{20} \\ p_{50} &= [12z(1,0) - 6z_u(1,0) + z_{uu}(1,0) - 12p_{00} - 6p_{10} - 2p_{20}]/2 \end{aligned} \tag{4.24}$$

Since p_{00}, p_{10} and p_{20} are already determined from (4.23), we can calculate p_{30}, p_{40} and p_{50} from (4.24).

Similarly, using the values of z, z_v and z_{vv} at P_3 (i.e., $u = 0$ and $v = 1$), we obtain

$$\begin{aligned} p_{03} &= [20z(0,1) - 8z_v(1,0) + z_{vv}(0,1) - 20p_{00} - 12p_{01} - 6p_{02}]/2 \\ p_{04} &= -15z(0,1) + 7z_v(0,1) - z_{vv}(0,1) + 15p_{00} + 8p_{01} + 3p_{02} \\ p_{05} &= [12z(0,1) - 6z_v(0,1) + z_{vv}(0,1) - 12p_{00} - 6p_{01} - 2p_{02}]/2 \end{aligned} \tag{4.25}$$

Substituting the values of p_{50} and p_{05} in (4.24) and (4.25), respectively, we obtain the values of p_{41} and p_{14} as

$$\begin{aligned} p_{41} &= \frac{5L_v \cos\theta_{uv}}{L_u} p_{50} \\ p_{14} &= \frac{5L_u \cos\theta_{uv}}{L_v} p_{05} \end{aligned} \tag{4.26}$$

Next, we use the values of z_v and z_{uv} at P_2 (i.e., $u = 1$ and $v = 0$) in (4.8) and obtain

$$\begin{aligned} p_{21} + p_{31} &= z_v(1,0) - p_{01} - p_{11} - p_{41} \\ 2p_{21} + 3p_{31} &= z_{uv}(1,0) - p_{11} - 4p_{41} \end{aligned}$$

Solving these equations, we obtain p_{21} and p_{31} as

$$\begin{aligned} p_{21} &= 3z_v(1,0) - z_{uv}(1,0) - 3p_{01} - 2p_{11} + p_{41} \\ p_{31} &= -2z_v(1,0) + z_{uv}(1,0) + 2p_{01} + p_{11} - 2p_{41} \end{aligned} \tag{4.27}$$

Similarly, using the values of z_u and z_{uv} at P_3 (i.e., $u = 0$ and $v = 1$) in (4.8), we obtain

$$\begin{aligned} p_{12} &= 3z_u(0,1) - z_{uv}(0,1) - 3p_{10} - 2p_{11} + p_{14} \\ p_{13} &= -2z_u(0,1) + z_{uv}(0,1) + 2p_{10} + p_{11} - 2p_{14} \end{aligned} \tag{4.28}$$

Rewritting (4.22) as

$$g_1 p_{32} + g_2 p_{23} = h_1 \tag{4.29}$$

where

$$\begin{aligned} g_1 &= A^2C(3BC + 2AD) \\ g_2 &= AC^2(2BC + 3AD) \\ h_1 &= -5A^4Bp_{50} - A^3(4BC + AD)p_{41} - C^3(BC + 4AD)p_{14} - 5C^4Dp_{05} \end{aligned} \tag{4.30}$$

with A, B, C and D as defined in (4.21). From the value of z_{vv} at P_2 in (4.8), we obtain

$$p_{22} + p_{32} = h_2 \tag{4.31}$$

where

$$h_2 = (1/2)z_{vv}(1,0) - p_{02} - p_{12} \tag{4.32}$$

Similarly, from the value of z_{uu} at P_3 in (4.8), we obtain

$$p_{22} + p_{23} = h_3 \tag{4.33}$$

where

$$h_3 = (1/2)z_{uu}(1,0) - p_{20} - p_{21} \tag{4.34}$$

Solving (4.29), (4.31) and (4.33), we obtain

$$\begin{aligned} p_{22} &= \frac{g_1 h_2 + g_2 h_3 - h_1}{g_1 + g_2} \\ p_{32} &= h_2 - p_{22} \\ p_{23} &= h_3 - p_{22} \end{aligned} \tag{4.35}$$

with g_1, g_2, h_1, h_2 and h_3 given by (4.30), (4.32) and (4.34).

Thus, with all the coefficients of the polynomial determined for a particular triangle, any point included inside or on the boundary of the triangle can be interpolated using the polynomial.

4.7 Wavelet-Based 2-D Interpolation

In section 3.5, a wavelet approach for interpolating 1-D signals was described. Analysis of 2-D systems can be performed in a similar fashion to their 1-D counterparts. If one uses the scheme as in Figure 4.10 then all 1-D results are easily extended to the 2-D case. The signal now is considered as a 1-D vector consisting of subsequent rows of the 2-D signal. The analysis matrix $\mathbf{W}$ can be formulated in a similar fashion as was done for the 1-D case. However, at each stage horizontal as well as vertical decompositions have to be considered. It can be noted that for a 2-D signal of size $N \times N$ the size of the analysis matrix $\mathbf{W}$ is $N^2 \times N^2$ and hence the transformation is computationaly very extensive.

We consider a 2-D signal in the form of a 4×8 matrix expressed as

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} & x_{17} & x_{18} \\ x_{21} & x_{22} & x_{22} & x_{24} & x_{25} & x_{26} & x_{27} & x_{28} \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} & x_{37} & x_{38} \\ x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} & x_{47} & x_{48} \end{bmatrix}$$

and convert it into a 1-D signal as

$$\hat{\mathbf{X}} = \begin{bmatrix} \hat{\mathbf{X}}_1 & \hat{\mathbf{X}}_2 & \hat{\mathbf{X}}_3 & \hat{\mathbf{X}}_4 \end{bmatrix}^T$$

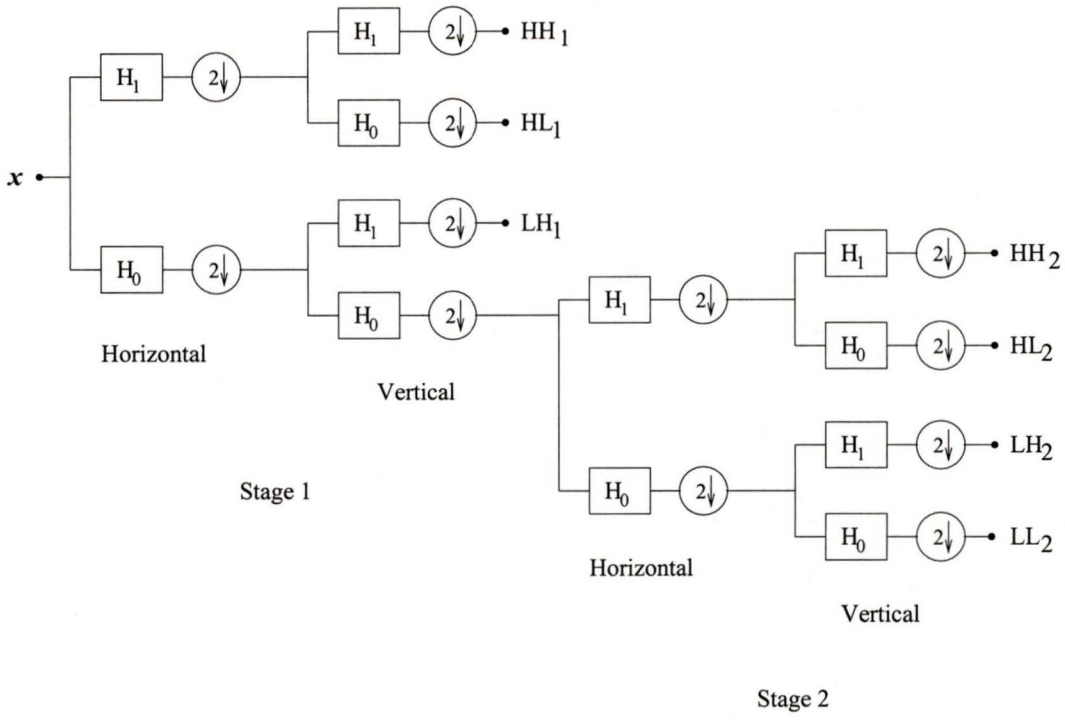


Figure 4.10. *Separable filter bank in two dimensions, with separable down-sampling by* 2.

where

$$\begin{aligned}
\hat{\mathbf{X}}_{\mathbf{1}} &\equiv [\, x_{11} \quad \cdots \quad x_{18} \,] \\
\hat{\mathbf{X}}_{\mathbf{2}} &\equiv [\, x_{21} \quad \cdots \quad x_{28} \,] \\
\hat{\mathbf{X}}_{\mathbf{3}} &\equiv [\, x_{31} \quad \cdots \quad x_{38} \,] \\
\hat{\mathbf{X}}_{\mathbf{4}} &\equiv [\, x_{41} \quad \cdots \quad x_{48} \,]
\end{aligned}$$

For a complete decomposition of $\mathbf{X}$ into its wavelet coefficients, we require a 2-stage filter bank as shown in Figure 4.10. Let $\mathbf{A}_2$, $\mathbf{A}_4$ and $\mathbf{A}_8$ denote the transformation matrices which convert a set of 2, 4 and 8 data points, respectively, into its wavelet coefficients using a single-stage filter-bank. For Haar filters they are given as

$$\mathbf{A}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\mathbf{A}_4 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

$$\mathbf{A}_8 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Let $\mathbf{Z}_2$, $\mathbf{Z}_4$ and $\mathbf{Z}_8$ denote null matrices of size 2×2, 4×4 and 8×8 respectively and $\mathbf{I}_2$, $\mathbf{I}_4$ and $\mathbf{I}_8$ denote identity matrices of size 2×2, 4×4 and 8×8, respectively. Let $\mathbf{Y}_h^1$ be the wavelet coefficient matrix after the horizontal decomposition of $\mathbf{X}$ in the first stage. $\mathbf{Y}_h^1$ is given by

$$\mathbf{Y}_h^1 = \mathbf{W}_h^1 \hat{\mathbf{X}} \tag{4.36}$$

where

$$\mathbf{W}_h^1 = \begin{bmatrix} \mathbf{A}_8 & \mathbf{Z}_8 & \mathbf{Z}_8 & \mathbf{Z}_8 \\ \mathbf{Z}_8 & \mathbf{A}_8 & \mathbf{Z}_8 & \mathbf{Z}_8 \\ \mathbf{Z}_8 & \mathbf{Z}_8 & \mathbf{A}_8 & \mathbf{Z}_8 \\ \mathbf{Z}_8 & \mathbf{Z}_8 & \mathbf{Z}_8 & \mathbf{A}_8 \end{bmatrix} \tag{4.37}$$

Here, $\mathbf{Y}_h^1$ contains the subsequent rows of the matrix formed by transforming $\mathbf{X}$ into its wavelet coefficients considering horizontal decomposition only.

Similarly, let $\mathbf{Y}_v^1$ be the wavelet coefficient matrix after the vertical decomposition of $\mathbf{Y}_h^1$ in the first stage. $\mathbf{Y}_v^1$ is given by

$$\mathbf{Y}^1 = \mathbf{Y}_v^1 = \mathbf{W}_v^1 \mathbf{Y}_h^1 \tag{4.38}$$

where

$$\mathbf{W}_v^1 = \begin{bmatrix} a_4^{11}\mathbf{I}_8 & a_4^{12}\mathbf{I}_8 & a_4^{13}\mathbf{I}_8 & a_4^{14}\mathbf{I}_8 \\ a_4^{21}\mathbf{I}_8 & a_4^{22}\mathbf{I}_8 & a_4^{23}\mathbf{I}_8 & a_4^{24}\mathbf{I}_8 \\ a_4^{31}\mathbf{I}_8 & a_4^{32}\mathbf{I}_8 & a_4^{33}\mathbf{I}_8 & a_4^{34}\mathbf{I}_8 \\ a_4^{41}\mathbf{I}_8 & a_4^{42}\mathbf{I}_8 & a_4^{43}\mathbf{I}_8 & a_4^{44}\mathbf{I}_8 \end{bmatrix} \tag{4.39}$$

In (4.39) a_4^{ij} denotes the element in the ith row and the jth column of $\mathbf{A}_4$. Therefore, the wavelet transformation matrix $\mathbf{W}^1$ which converts $\mathbf{X}$ into its wavelet coefficients after stage one is given by

$$\mathbf{W}^1 = \mathbf{W}_v^1 \mathbf{W}_h^1 = \begin{bmatrix} a_4^{11}\mathbf{A}_8 & a_4^{12}\mathbf{A}_8 & a_4^{13}\mathbf{A}_8 & a_4^{14}\mathbf{A}_8 \\ a_4^{21}\mathbf{A}_8 & a_4^{22}\mathbf{A}_8 & a_4^{23}\mathbf{A}_8 & a_4^{24}\mathbf{A}_8 \\ a_4^{31}\mathbf{A}_8 & a_4^{32}\mathbf{A}_8 & a_4^{33}\mathbf{A}_8 & a_4^{34}\mathbf{A}_8 \\ a_4^{41}\mathbf{A}_8 & a_4^{42}\mathbf{A}_8 & a_4^{43}\mathbf{A}_8 & a_4^{44}\mathbf{A}_8 \end{bmatrix} \tag{4.40}$$

After careful consideration, the wavelet transformation matrix $\mathbf{W}$ which converts $\mathbf{X}$ into its wavelet coefficients after two stages is given by

$$\mathbf{W} = \mathbf{W}^2 \mathbf{W}^1 \tag{4.41}$$

where $\mathbf{W}^2$ which converts $\mathbf{W}^1$ into its wavelet coefficients is given by

$$\mathbf{W}^2 = \begin{bmatrix} a_2^{11}\mathbf{A}_4 & \mathbf{Z}_4 & a_2^{12}\mathbf{A}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 \\ \mathbf{Z}_4 & \mathbf{I}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 \\ a_2^{11}\mathbf{A}_4 & \mathbf{Z}_4 & a_2^{12}\mathbf{A}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 \\ \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{I}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 \\ \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{I}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 \\ \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{I}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 \\ \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{I}_4 & \mathbf{Z}_4 \\ \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{Z}_4 & \mathbf{I}_4 \end{bmatrix} \tag{4.42}$$

In (4.42) a_2^{ij} denotes the element in the ith row and the jth column of $\mathbf{A}_2$. Thus

$$\mathbf{Y}^2 = \mathbf{W}\,\hat{\mathbf{X}} \tag{4.43}$$

where $\mathbf{Y}^2$ contains the subsequent rows of the matrix formed by transforming $\mathbf{X}$ into its wavelet coefficients considering a two-stage decomposition. The wavelet transform matrix $\mathbf{W}$ can be similarly determined for a data of a larger size.

Once the wavelet transformation matrix $\mathbf{W}$ is formulated, data can be interpolated in the Sobolev space in the same way as was done for a 1-D signal which has been discussed in section 3.4.2. The inverse wavelet transform of $\hat{\mathbf{w}}$ will produce, in the time-domain, the interpolated data in the form of subsequent rows of the estimated 2-D signal.

4.8 Interpolation Results and Discussion

We consider a 2-D signal which has both smooth as well as sharp edges, as shown in Figure 4.11

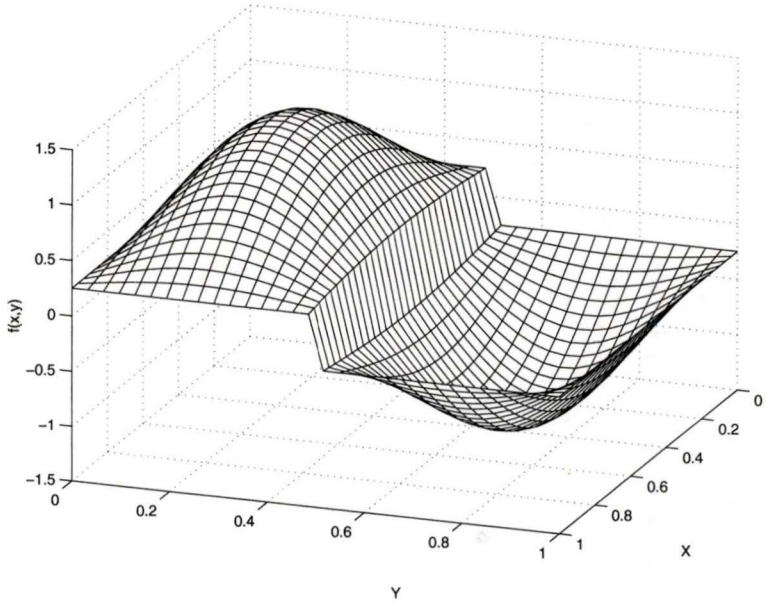


Figure 4.11. *Original signal.*

To illustrate the behavior of the interpolation techniques, we arbitrarily sample 300 data points from the original 2-D signal as shown in Figure 4.12. In order to estimate the original signal from the data samples data is interpolated at a regular rectangular grid. The sampled data is interpolated using both the triangle-based method as well as the wavelet based method.

The triangle-based interpolation method starts by considering a *super-triangle* with vertices at (0.5,5), (-2.5,-3) and (3.5,-3). As the x and y coordinates of the

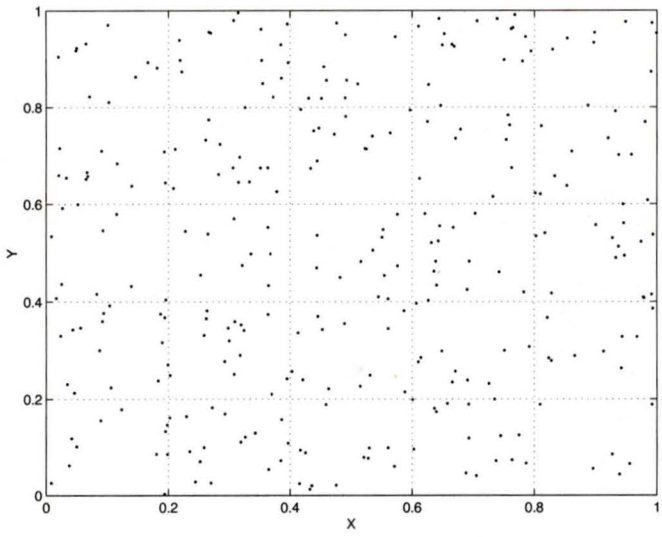


Figure 4.12. 300 *sample points of the original signal.*

data points have initially normalized to a scale of [0 1], the window enclosing all the data points is well within the *super-triangle*. The data points are inserted into the triangulation one at a time and the triangulation updated. When all the 300 data points have been inserted, the triangles having one or more of the vertices of the *super-triangle* are removed from the triangulation. The final Delaunay triangulation is shown in Figure 4.13.

Once the Delaunay triangulation is complete, a data point inside a particular triangle is interpolated using a fifth-degree polynomial formed inside the triangle as discussed in section 4.4. A surface is thus formed by interpolating data points on a regular rectangular grid. The interpolated surface is shown in Figure 4.14. Although this is a good estimate of the original data, both at smooth areas and sharp edges, it can be observed that some interpolated data points, especially at the boundary, do not conform to the local characteristics of the data. This can be attributed to the fact that the Delaunay triangulation formed as shown in Figure 4.13 has some thin triangles, especially at the edges. As polynomials formed inside thin triangles do not produce good interpolation results they are unsuitable for interpolation.

Next, the same set of sample points is used to interpolate a surface using the wavelet-based interpolation process described earlier. Initially the data points are converted to dyadic points by approximating them to be placed at binary rationals.

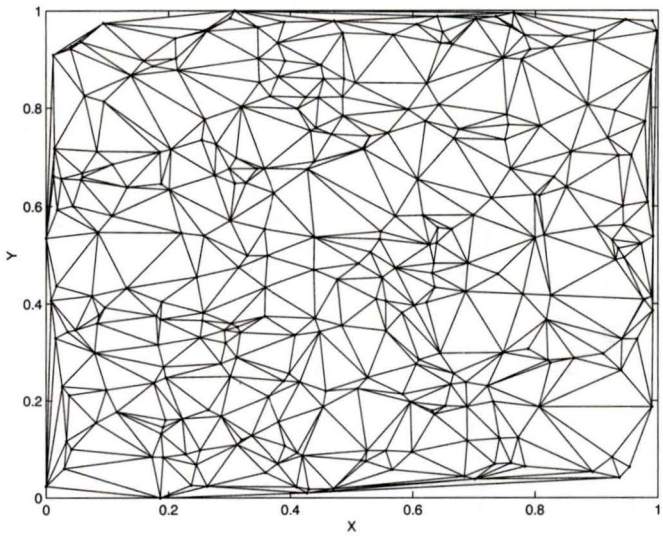


Figure 4.13. *Delaunay triangulation of the sampled data points.*

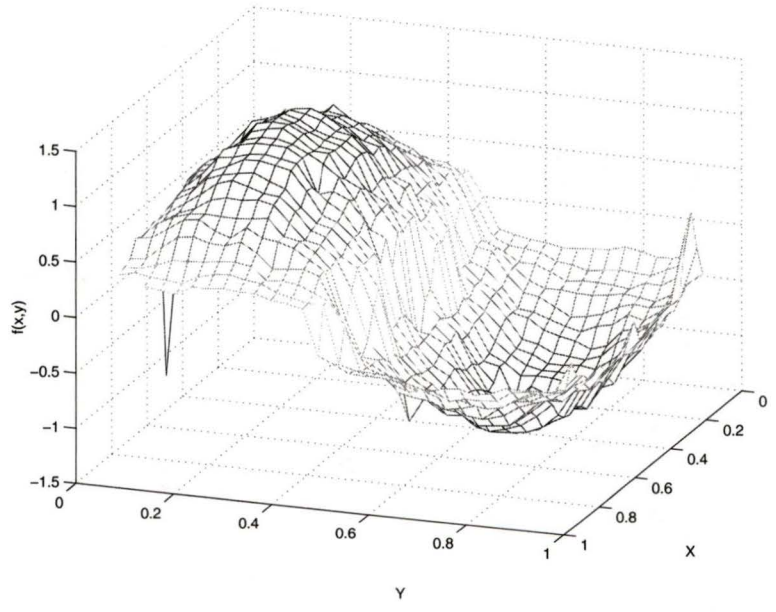


Figure 4.14. *Surface interpolated using the triangulation method.*

Considering that the final interpolated data is of size 32×32, data points are placed at $(k/2^5, l/2^5)$, $1 \leq k \leq 32$, $1 \leq l \leq 32$ for arbitrary integers k and l. It can be noted that two or more data points can be approximated at the same dyadic point.

In such case an average of the elevation values of all such data points is considered as the elevation value at that point. Figure 4.15 shows the dyadic points, which have been approximated from the sample data points shown in Figure 4.12. Thus data is

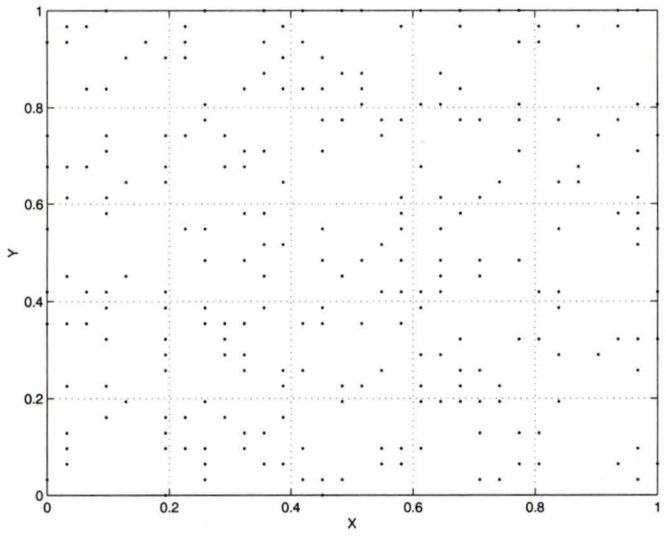


Figure 4.15. *Dyadic points, approximated from the sampled data points.*

placed at a regular grid with missing points in between. The missing points are then interpolated using the wavelet-based interpolation technique.

The resulting surface is shown in Figure 4.16. It can be observed in Figure 4.16 that the wavelet-based interpolation technique gives fairly good results for regions which are smooth in nature. However, it tries to smoothen regions with sharp edges if a sufficient number of data points is not available at places near the sharp edges. The root mean square errors between the original data and the interpolated data using the triangle based method and the wavelet based method are found out to be 0.0655 and 0.0612 respectively.

Next, we apply the two interpolation techniques to obtain the ground and canopy profiles for a scanning LIDAR data. After initially categorizing the data points as ground hits or canopy hits, the scattered data points are interpolated to obtain the corresponding profiles.

Figure 4.17 shows the ground profile obtained by interpolating the ground hits using the triangle-based interpolation process whereas Figure 4.18 shows the ground profile obtained by interpolating the data points categorized as ground hits using the

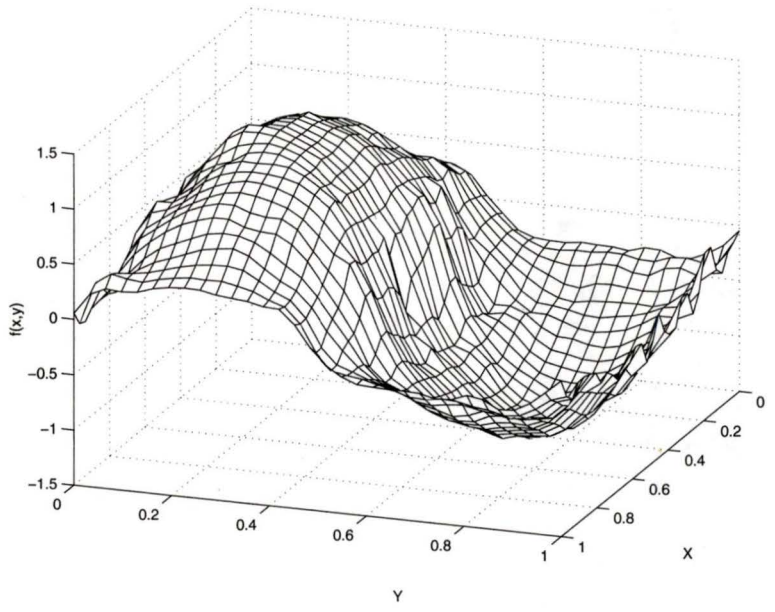


Figure 4.16. *Surface interpolated using the wavelet-based method.*

wavelet-based interpolation process. The ground profile indicates a hilly area and is fairly smooth in nature barring some regions with uneven surface charateristics, as seen from Figure 4.17. Possible reasons for this unevenness can be the presence of rocks or houses. Another possibility may be due to false categorization of data points at these regions. This may happen if the neighborhood of the data point to be categorized does not include any ground hit and all the data points are vegetation hits. In that case, a data point under consideration will be categorized as a ground hit even though it is a vegetation hit. Thus at places where the vegetation hits are very dense the neighborhood search area should be sufficiently large so that it may contain at least a single ground hit. On the other hand, the ground profile obtained with the wavelet-based interpolation process for the same set of data points has occasional uneven regions and undulated surface patches especially at places where the density of the data points is very low, as seen in Figure 4.18. The surface in this case is fitted considering a Sobolev space and may not exactly represent the actual surface.

Figures 4.19 and 4.20 show the canopy profiles obtained by interpolating the data points categorized as canopy hits using the triangle-based method and the wavelet-based method, respectively. Here, both the profiles obtained show similar surface

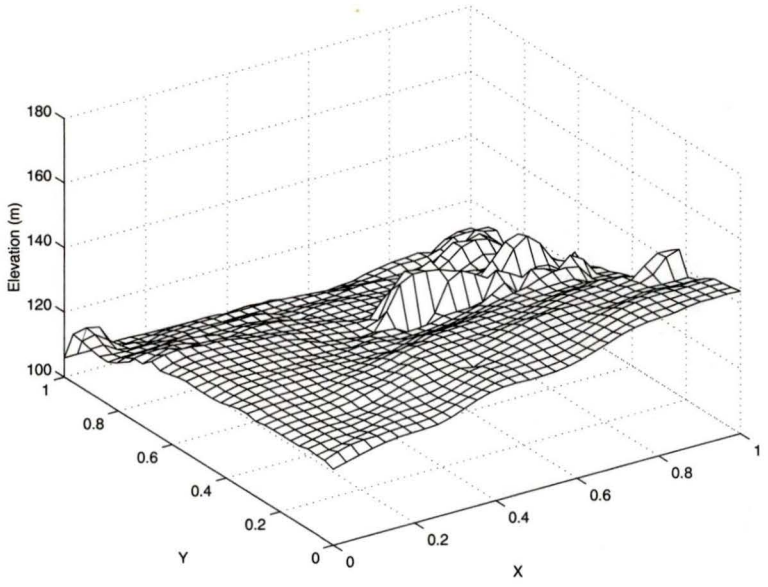


Figure 4.17. *Ground profile obtained using the triangle-based interpolation process.*

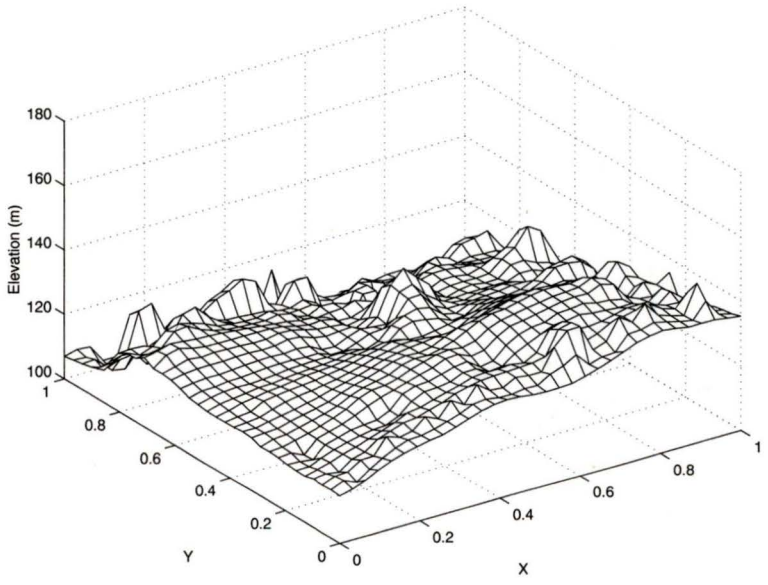


Figure 4.18. *Ground profile obtained using the wavelet-based interpolation process.*

characteristics.

We can conclude that a large part of the region is covered with vegetation, presumably comprising trees, as can be inferred by comparing the elevation in the canopy

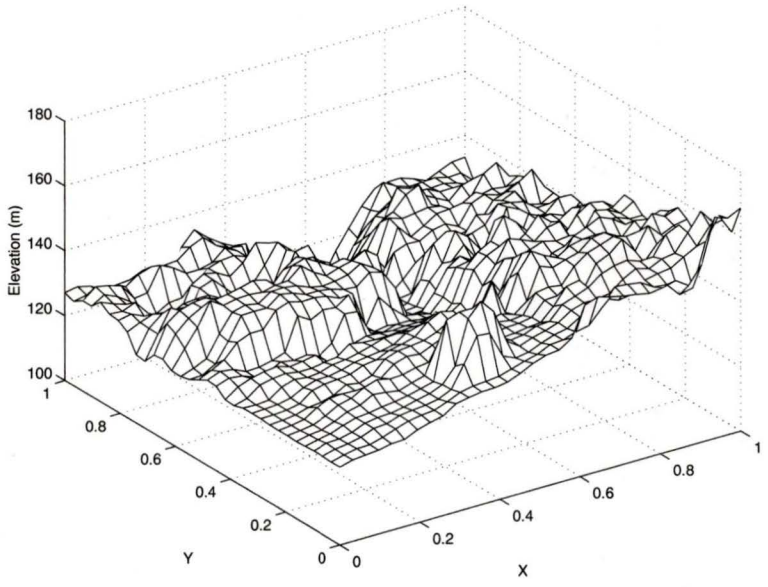


Figure 4.19. *Canopy profile obtained using the triangle-based interpolation process.*

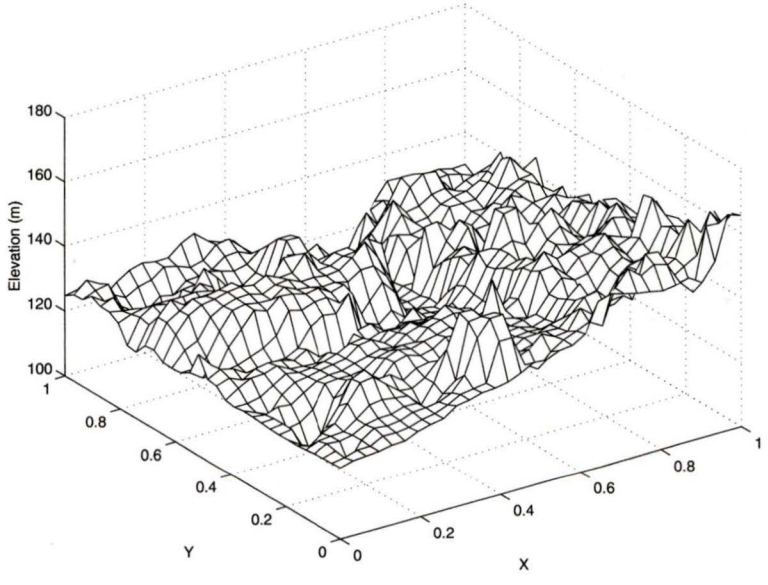


Figure 4.20. *Canopy profile obtained using the wavelet-based interpolation process.*

profile with that in the ground profile. It is noted that at places where there is no vegetation the canopy coincides with the ground.

4.9 Conclusions

The problem of interpolating scattered 1-D data was examined in chapter 3. Chapter 4 extends the interpolating problem to 2-D. Two techniques for the purpose of interpolating data points scattered irregularly on a 2-D profile have been discussed. The first technique involves approximating the surface over a triangulated region. A Delaunay triangulation is constructed with the scattered data points as the vertices of the triangles. Data points are interpolated using a fifth-degree polynomial fitted over each triangular cell. The second technique builds on the wavelet-based 1-D interpolation process discussed in chapter 3. The wavelet transformation matrix, which converts a 2-D signal to its wavelet coefficients, has been formulated. This is followed by minimizing the Sobolev norm of the reconstructed signal which in the wavelet domain reduces to weighted least-squares optimization.

The interpolation techniques have been applied to obtain the ground and canopy profiles of a LIDAR scanning data. It was found that at places where the density of data is low, the triangle-based method of interpolation yields better results as compared to the wavelet-based interpolation process. However, when a thin triangle is present in the triangular network, the polynomial fitted into it experiences oscillatory behavior and produces incorrect interpolation values. On the other hand, at places where the density of the data is quite high the wavelet-based process yields better results. The computational cost of the wavelet-based interpolation process is higher than the triangle-based process and increases significantly with increasing size of the data.

Chapter 5

Delineation of the Transmission Wire

5.1 Introduction

When a transmission wire is supported by poles, the part of the wire from one pole to the next forms a catenary, a curve formed by a hanging flexible wire supported at its ends and acted upon by a uniform gravitational force. Thus, in order to delineate the transmission wire it is imperative to approximate the wire hits with a series of catenaries.

Data points that were classified earlier as wire hits can include laser hits reflected from poles supporting the wire, tall trees or birds. In addition the data may also include impulsive noise values. In order to delineate the catenary wires laser hits actually reflected from the wires have to be identified.

We consider the fact that the projection of a catenary wire from one pole to the next on the x-y plane forms a straight line. Thus the data points whose projection on the x-y plane doesn't lie on a straight line can be discarded. Although this may remove some data points which are not wire hits, the data set may still contain some data points not reflected from the wire but has its x-y projection aligned with the line of the transmission wire. In order to get rid of these unwanted data points and to obtain a smooth curve representing the wire hits are approximated by a chain of catenaries using an optimality criterion.

5.2 Tracing the Wire

A transmission wire is initially traced by collecting all the data points, among the set of points classified as wire hits, whose projection on the x-y plane maintains a chain of line segments. These data points are grouped in lists, each list representing a transmission wire. A list starts by considering a data point p and searching for its neighboring data points in order to find points collinear with p. If such points exist, then they can be part of the wire and so it is attempted to extend from both ends by inserting more data points collinear to the line. When the line can not be extended any further a check is made on the length of the line; if the length is too short to be a segment of a transmission wire the list is deleted and the process is repeated. Otherwise, the line is accepted as a segment of the transmission wire from one pole to the next or from a pole to one end of the data set. Next, an attempt is made to further extend the segment considering the fact that at a pole the transmission wire may take a course with a different angle. This is done by searching points which may be collinear with the point at the extreme end of the initial line. If such points exist then the second line is initiated. This line is similarly extended until it cannot be extended any further. The process is repeated until the series of line segments cannot be drawn any further. This completes the tracing of a transmission wire and the data set is then searched for more wires.

The process can be illustrated in a simple algorithm as given below. Each data point is assigned a status: *not_visited* - data point not yet considered, *visited* - considered but not listed as a wire hit and *listed* - data point included in a list of wire hits. Initially all data points are assigned the status *not_visited.*

1. Consider a data point p_i which is *not_visited.*
2. Find n nearest points of p_i which are *not_visited* and arrange them in increasing order of their distance from p_i.
3. Among the n nearest neighbors of p_i, find two points p_b and p_f such that p_{b1}, p_i and p_{f1} form a line, with an accepted angle of curvature with p_i at the middle, making an angle θ with the x-axis.

   If no such points exist then:

   Declare p_i as *visited.*

Consider another point which is *not_visited* and go to step 1.

Else continue.

4. Try to extend the line in the forward direction:
   a. Find n nearest points of p_{f1} which are *not_visited.*
   b. Find a point p_{f2} among the n nearest neighbors such that the line segment p_{f1}-p_{f2} maintains the same angle θ.
   c. If there exists such a point then:

   Include point p_{f2} in the line.

   $p_{f1} \leftarrow p_{f2}$.

   Go to step 4(a).

   Else continue. (The line cannot be extended any further for this angle.)
5. Try to extend the line in the reverse direction:
   a. Find n nearest points of p_{b1} which are not listed or visited.
   b. Find a point p_{b2} among the n nearest neighbors such that the line segment p_{b2}-p_{b1} maintains the same angle.
   c. If there exists such a point then:

   Include point p_{b2} in the line.

   $p_{b1} \leftarrow p_{b2}$.

   Go to step 5(a).

   Else continue. (The line cannot be extended any further for this angle.)
6. Calculate the length of the line, *len.*
7. If *len* < *chklen* then: (The line at this angle cannot be considered as a transmission wire.)

   Discard the points included in the line.

   Find a different combination of p_{b1} and p_{f1} with p_i in step 3 and proceed from step 3.

   Else declare all points in the line as listed.
8. Try to extend the line with a different angle in the forward direction:
   a. Find n nearest points of p_{f1} not listed or visited.

b. Among the n nearest points find two points p_{f2} and p_{f3} such that p_{f1}, p_{f2} and p_{f3} form a line with an accepted angle of curvature, p_{f2} being at the middle, making an angle β with the x axis.

c. If two such points exist then:

Include p_{f2} and p_{f3} in the line.

$p_{f1} \leftarrow p_{f3}$.

Do step 4.

Else continue.

9. Calculate the length of the line, if any, at the new angle and call it *len*.
10. If $len < chklen$ then:

Discard the points included in the line at the new angle.

Find a different combination of p_{f2} and p_{f3} with p_{f1} in step 8(b) and proceed from step 8(b).

If no such combination exists then go to step 11. (The line cannot be extended in the forward direction at any angle.)

Else

Declare all points in the line as listed.

Go to 8(a).

11. Try to extend the line with a different angle in the reverse direction:

a. Find n nearest points of p_{b1} not listed or visited.

b. Among the n nearest points find two points p_{b2} and p_{b3} such that p_{b1}, p_{b2} and p_{b3} form a line with an accepted angle of curvature, p_{b2} being at the middle.

c. If two such points exist then:

Include p_{b2} and p_{b3} in the line.

$p_{b1} \leftarrow p_{b3}$.

Do step 5.

Else continue.

12. Calculate the length of the line at the new angle, *len*.
13. If $len < chklen$ then:

Discard the points included in the line at the new angle.

Find a different combination of p_{b2} and p_{b3} with p_{b1} in step 11(b) and proceed from step 11(b).

If no such combination exists then go to step 14. (The line cannot be extended in the backward direction at any angle.)

Else

Declare all points in the line as listed.

Go to 11(a).

14. Repeat the whole process to look for other wires.

This completes the tracing of a transmission wire. As more than one transmission wire can exist, the whole process is repeated to trace other possible wires. The process is repeated until all the data points have either been visited or listed.

Due to unavailability of real data, the algorithm has been used over synthetic data. Figure 5.1 shows a set of possible wire hits. As mentioned earlier this may include hits from tall trees, birds or the poles supporting the wires. Figure 5.2 shows the traces of two parallel wires. It also shows two other smaller traces of data points whose projection on the x-y plane form straight lines. These two smaller traces are, however, not considered as the length of the traces indicate that they cannot be wires. This leaves us with two lists of data points corresponding to two wires.

The lists thus formed may include data points which are actually not wire hits but whose projection on the x-y plane aligns with the line which a wire projects on the x-y plane. In order to get rid of these unwanted data points from the lists, a simple algorithm, as given below, has been used.

1. Consider a point p in the list.
2. Consider $2n$ points, n on the left and n on the right of p.
3. Take the average, av, of the elevation values of the $2n$ number of points.
4. If the elevation value of p is greater or lesser than av by a certain threshold t then declare p as erroneous and replace the elevation value of p with av.
5. Repeat steps 1 - 4 for all points in the list.

Figure 5.3 shows possible bird hits or erroneous data points whose x-y projection aligns with the line the wire projects on the plane. Using this simple algorithm

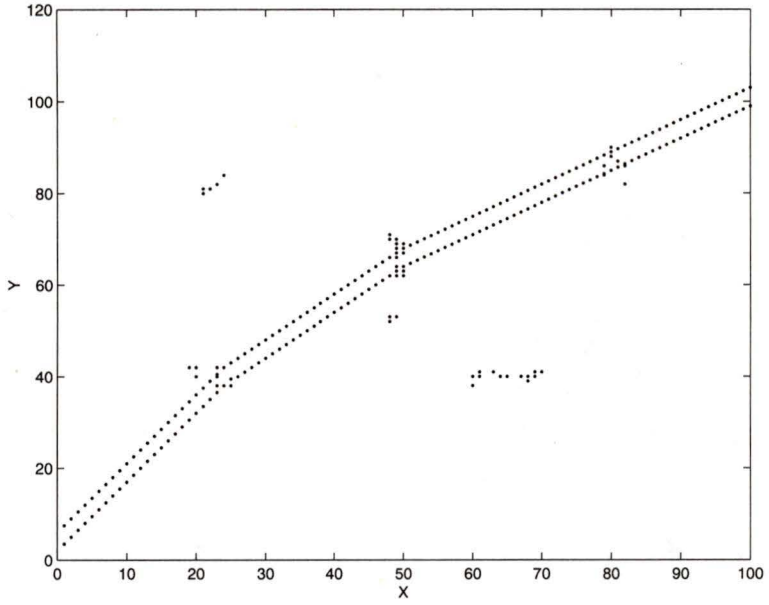


Figure 5.1. *Possible wire hits.*

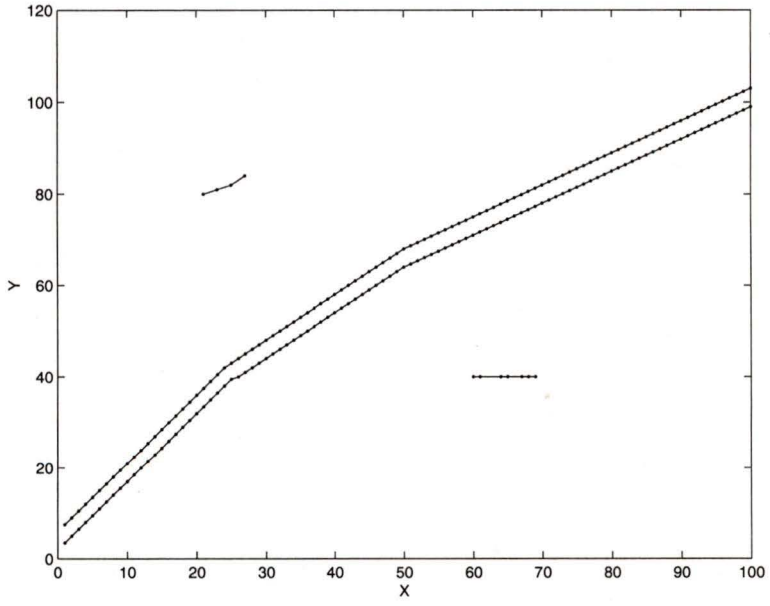


Figure 5.2. *Data points joined to trace possible catenary wires.*

the elevations of these unwanted data points have been approximated with suitable values as shown in Figure 5.4. Figure 5.5 shows the two transmission wires and their

projection on the x-y plane.

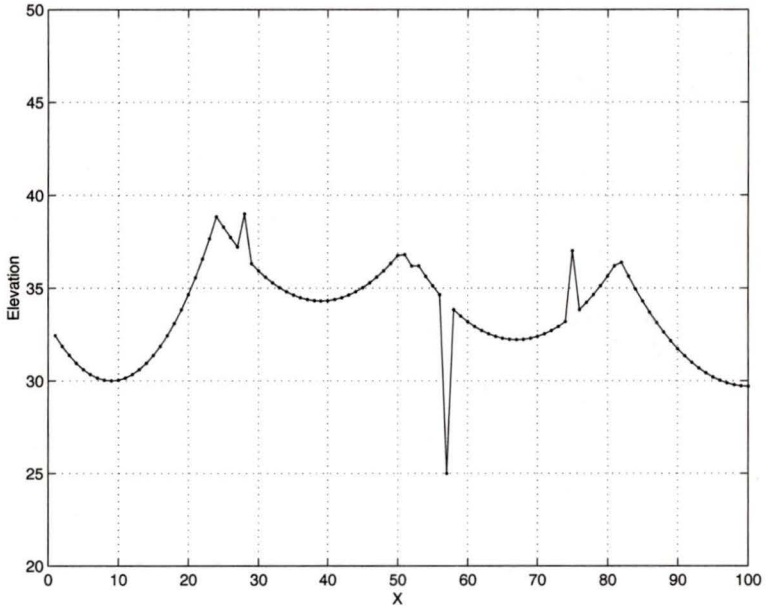


Figure 5.3. *Trace of a wire with erroneous data points.*

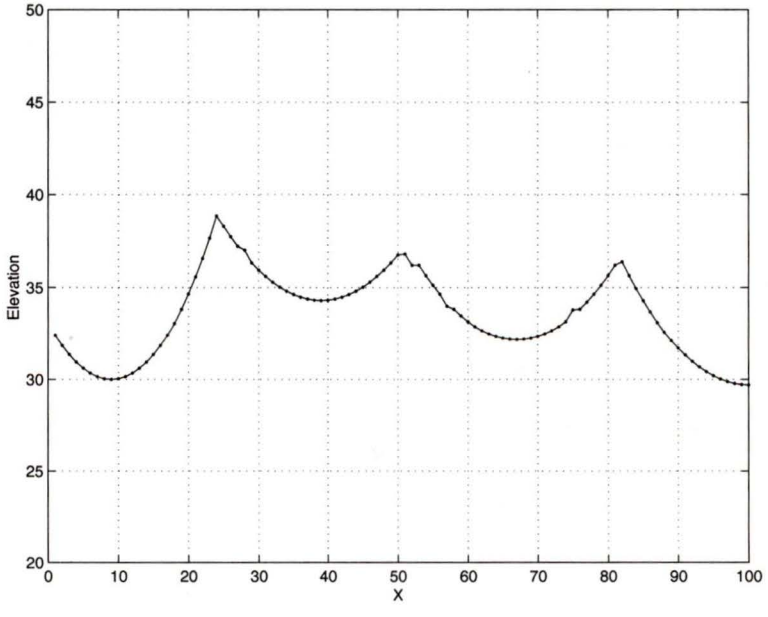


Figure 5.4. *Unwanted data points replaced with suitable values.*

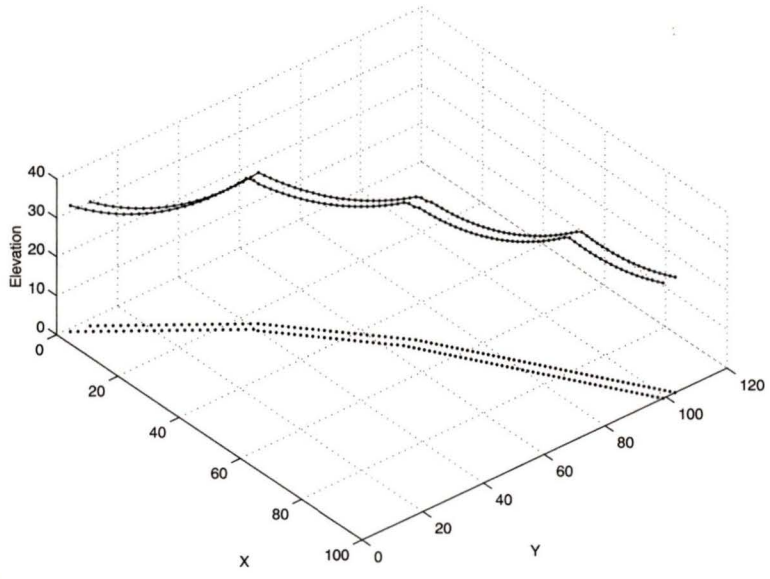


Figure 5.5. *Transmission wires and their projection on the x-y plane.*

In order to approximate the wire hits with a series of catenaries, two poles being the ends of each catenary, the poles have to be first marked. The following algorithm marks the pole positions for the lists, each list representing a transmission wire. The algorithm starts by considering three moving windows: $left, middle$ and $right$. The windows contain consecutive data points in the list, each containing a fixed number of points. The data point having the maximum elevation value in each window is noted. If the elevation value of the data point having the maximum elevation in the $middle$ window, say p, is greater than the corresponding values in the $left$ and the $right$ windows then p is marked as the pole. The windows are moved one window length and the process is repeated until the windows have covered the whole wire length.

1. Consider the first n data points in the list as contained in $left$ window, the next n in $middle$ window and the n next in $right$ window.
2. Set max_l as the data point having the maximum elevation among the n data points in the $left$ window.
3. Similarly set max_m and max_r as the data points having the maximum elevation value in the $middle$ and the $right$ windows, respectively.

4. If the elevation value of *max_middle* is greater than that of both max_l and max_r then declare max_m as a pole position. Otherwise, shift the windows towards right by a window length, i.e., n data points and repeat steps 2, 3 and 4 until all data points have been included.
5. Consider the first and last data points in the list also as pole positions. (This ensures that the first and the last catenary curves are not omitted out.)

Once the poles have been marked, a catenary curve can be drawn between two consecutive poles, which best fits the wire hits. All the catenary curves can then be joined to form the transmission wire.

5.3 Optimizing the Catenary Curve

As a transmission wire forms a series of catenaries from pole to pole, it is imperative to approximate the wire hits with a series of catenaries. The equation of a catenary from a pole to the next can be formulated as

$$y(x) = b\cosh\left(\frac{x - x_0}{a}\right) - y_0 \tag{5.1}$$

where x_0 is the point where the catenary is at its lowest elevation having a value y_0, a and b being constants.

Let $\mathbf{d} = [a\ b\ x_0\ y_0]^T$. Then equation (5.1) can be written as

$$\mathbf{Y}(\mathbf{d}, x) = b\cosh\left(\frac{x - x_0}{a}\right) - y_0 \tag{5.2}$$

Let the wire hits from a pole to the next be represented by $\mathbf{Y}_0(x_k)$, where, x_k, $k = 1, 2, \ldots, K$ be the locations of the wire hits. An approximation error can, therefore, be expressed as

$$e(\mathbf{d}, x_k) = \mathbf{Y}(\mathbf{d}, x_k) - \mathbf{Y}_0(x_k) \tag{5.3}$$

This can be formed as a column vector

$$E(\mathbf{d}) = [e_1(\mathbf{d})\ \ e_2(\mathbf{d})\ \ \ldots\ \ e_k(\mathbf{d})]^T \tag{5.4}$$

where

$$e_i(\mathbf{d}) = e(\mathbf{d}, x_i) \tag{5.5}$$

for $k = 1, 2, \ldots, K$. An objective function can be defined in terms of the L_p norm of $E(\mathbf{d})$ as

$$\Psi(\mathbf{d}) = L_p = \left[\sum_{i=1}^{K} |e_i(\mathbf{d})|^p\right]^{1/p} \tag{5.6}$$

where p is an integer. The magnitude of the largest amplitude-response error is usually required to be as small as possible and, therefore, the L_∞ norm of the error function should be used. Algorithms developed specifically for the minimization of the L_∞ norm are known as *minimax* algorithms and lead to designs in which the error is uniformly distributed with respect to an independent variable.

A common minimax algorithm is the *least-pth* algorithm which involves minimizing an objective function of the type given is (5.6) for increasing values of p, say $p = 2, 4, 8, \ldots$, etc., and is as follows

1. Input $\breve{\mathbf{d}}_0$ and ϵ_1. Set $j = 1$, $p = 2$, $\mu = 2$, $\hat{E}_0 = 10^{99}$.
2. Initialize locations $x_1, x_2, \ldots, x_K$.
3. Using $\breve{\mathbf{d}}_{j-1}$ as initial value, minimize

$$\Psi_j(\mathbf{d}) = \hat{E}(\mathbf{d}) \left\{\sum_{i=1}^{K} \left[\frac{|e_i(\mathbf{d})|}{\hat{E}(\mathbf{d})}\right]^p\right\}^{1/p}$$

where

$$\hat{E}(\mathbf{d}) = \max_{1 \le i \le K} |e_i(\mathbf{d})|$$

with respect to $\mathbf{d}$, to obtain $\breve{\mathbf{d}}_j$. Set $\hat{E}_j = \hat{E}(\breve{\mathbf{d}})$.

4. If $|\hat{E}_{j-1} - \hat{E}_j| < \epsilon_1$, then output $\breve{\mathbf{d}}_j$ and $\hat{E}_j$, and stop. Else set $p = \mu p$, $j = j + 1$ and go to step 3.

The minimization in step 3 can be carried out by using a standard optimization method such as the quasi-Newton optimization method. The gradient of $\Psi_j(\mathbf{d})$ is given as

$$\nabla \phi(\mathbf{d}) = \left\{\sum_{i=1}^{K} |e_i(\mathbf{d})|^2\right\}^{-1/2} \sum_{i=1}^{K} |e_i(\mathbf{d})| \nabla |e_i(\mathbf{d})| \tag{5.7}$$

where

$$\nabla |e_i(\mathbf{d})| = \begin{cases} \nabla e_i(\mathbf{d}), & e_i(\mathbf{d}) \ge 0 \\ -\nabla e_i(\mathbf{d}), & \text{otherwise} \end{cases} \tag{5.8}$$

Thus, $\mathbf{Y}[\breve{\mathbf{d}}_j, x_k]$ produces the best fit catenary curve. Although the optimization process yields smooth catenary curves representing the transmission wires, it is computationally very expensive and time consuming. Due to these reasons the optimization process has not been used.

5.4 Conclusions

Several algorithms used to delineate transmission wires have been discussed in this chapter. The process starts by grouping data points which are expected to be actual wire hits into lists, each list representing a transmission wire, from a set of data points classified earlier as possible wire hits. This is done keeping in mind that transmission wires project straight lines on the x-y plane. Next, data points which have been included in the lists representing the wires but in fact are not wire hits, are eliminated. The process has been tried on synthetic data as real data were unavailable. Although simple algorithms have been used for the purpose, complications may arise with real data and the algorithms used may have to be modified accordingly for better performance. The chapter also presents a method of fitting a smooth catenary curve to the wire hits by using an optimization process.

Chapter 6

Conclusions and Scope of Future Work

The thesis is based on a project sponsored by Terra Remote Sensing Inc., for the development of efficient data processing techniques to create a topological map of an area surveyed by a scanning laser system. A light airplane with a scanning laser system installed has been used to acquire data. Elevation values of the region under survey have been calculated from the laser hits reflected from the topological surface. Initially data have been acquired by a laser profiling system but since a laser profiling system cannot survey a large area at one time, Terra Remote Sensing Inc. has developed a laser scanning system to acquire topological data.

In order to obtain the ground and canopy profiles and to delineate transmission wires, laser hits have been classified as ground hits, vegetation hits or wire hits. This has been done using a method based on order statistics filter as described in chapter 2. Some preliminary processing techniques have been implemented so as to make the processing of data simple and efficient. The algorithm developed for searching the nearest neighbors of a data point greatly reduces the time for the classification of the laser hits. This algorithm has also been used in the interpolation technique described in chapter 4.

The data points classified as ground, vegetation or wire hits are scattered. In order to obtain the ground and canopy profiles, these scattered data points have been interpolated. Moreover, as a profiler system acquires data which can be considered 1-D, the interpolation technique used for this data is also 1-D. Two techniques have been used to interpolate 1-D profiler data. The first is based on the Deslauriers-Dubuc (DD) process, which is a multiscale method of iteratively interpolating uniformly

sampled data points. The data points are initially approximated in terms of binary rationals. The DD process interpolates data halfway between given data points by using polynomial interpolation with the help of neighboring data points. The actual nonuniform interpolation process used has been considered to be a two-stage process. In the first stage, nonuniform data have been regularized by iteratively interpolating data points in between the given data points in a way somewhat similar to the DD process. Once this has been done, the uniform data is further interpolated, at the second stage, using the DD process to obtain a smooth curve.

The second method used is a wavelet approach to nonuniform interpolation. Here a curve is constructed from the given data points using an optimality criterion. The optimality criterion choosen is the Sobolev norm of the constructed curve which has been minimized so as to maximize the smoothness of the constructed curve.

Ground and canopy profiles have been obtained using both of the techniques. The wavelet-based method yields better results in the sense that the constructed profiles are smoother than those obtained using the iterative technique, especially when the scattered data points are sparsely populated. However, the computational cost of the wavelet-based approach is significantly higher than that of the iterative process. The iterative process is more suited when the data points are fairly dense and evenly situated.

Unlike the profiling system, the laser mounted in the scanning system scans a large part of the survey area at one time along the flightline. Hence 2-D interpolation has been used to obtain 3-D profiles from data acquired by the scanning laser system. Two techniques have been implemented for interpolating 2-D scattered data. The first method is based on approximating surfaces over a triangulated network. An optimal triangulated network, the Delaunay triangulation, has been used to ensure that the triangles formed are as close to equiangular triangles as possible. Once the Delaunay triangulation has been formed, a bivariate fifth-degree polynomial is fitted in each of the triangles. The interpolation function is smooth and its first-order partial derivatives are continuous on the boundaries of the triangle.

The second method of 2-D interpolation is similar to the wavelet approach of interpolation used for interpolating the 1-D profiler data. Here, the 2-D data have been considered as a 1-D signal which contains subsequent rows of the 2-D signal. A

2-D signal is reconstructed whose Sobolev norm has been minimized. In the wavelet domain, this reduced to a weighted least-squares optimization problem.

Two techniques have been used to obtain 3-D ground and canopy profiles. The triangle-based interpolation process yielded smoother profiles compared to the wavelet-based approach at places where the data is sparse. Moreover, the wavelet-based method has a very high computational cost which increases significantly with the size of the data. However, often some of the triangles formed in the Delaunay triangulation, especially the ones at the border of the data, are thin which produces inaccurate interpolation results.

Algorithms have been developed to delineate the transmission wires. The process starts by tracing the wire by grouping the data points which are most likely to be wire hits from the set of data points classified earlier as possible wire hits. This is done considering the fact that transmission wires project straight lines on the x-y plane. An optimization method has been presented to fit smooth catenary curves to the wire hits. Synthetic data have been used for the process as real data was unavailable. With real data, complications may arise and there may be the need to modify the algorithms.

The work done so far provides a base for further research into the various aspects of the project. The algorithm used for the classification of the data points is quite simple and straightforward. The data density and spatial corelation between the data points need to be looked into in order to classify data points more accurately. Spatial aliasing should be considered while sampling a known signal in order to better analyze the reconstructed signal after applying the interpolation processes. The data points have been converted to their wavelet coefficients in the wavelet-based interpolation process. This conversion process is quite cumbersome and time-consuming. A faster process will greatly enhance the efficiency of the interpolation process. A 2-D data set has been first converted to a 1-D signal consisting of the subsequent rows of the 2-D data before converting it to its corresponding wavelet coefficients. A better way to do this would be to directly convert the 2-D data to its corresponding wavelet coefficients without breaking it into a 1-D signal although this will highly increase the complexity of the process. A few improvements have been recently proposed by Sloan for the Delaunay triangulation method which can be implemented for a better performance

of the triangulation process. Finally a lot of improvements have to be made for the process of delineating the transmission wires. Depending on the actual wire data, adjustments have to be made for the classification of the wire hits. Special provisions have to be taken for the presence of multiple transmission wires. The optimization process to draw a smooth curve representing the wires should be investigated further with real-life data and methods of reducing the computational complexity should be explored.

Bibliography

[1] G. D. Hickman and J. E. Hogg, "Application of an airborne pulsed laser for near shore bathymetry measurements," *Remote Sensing of Environment*, vol. 1, pp. 47-58, 1969.

[2] W. Rattman and T. Smith, "Lasers for depth sounding and underwater viewing," *Hydrospace*, vol. 5, no. 1, pp. 57-59, Feb. 1972.

[3] C. Bressel, I. Itzkan, J. E. Nunes and F. Hoge, "Airborne oceanographic lidar system," *Proceedings of the 11th International Symposium on Remote Sensing of Environment*, vol. II, pp. 1259-1269, 1977.

[4] F. E. Hoge, R. N. Swift and E. B. Frederick, "Water depth measurement using an airborne pulsed neon laser system," *Applied Optics*, vol. 19, pp. 871-883, 1980.

[5] L. E. Link and J. G. Collins, "Airborne laser systems use in terrain mapping," *Proceedings of the 15th International Symposium on Remote Sensing of Environment*, pp. 95-110, 1981.

[6] W. B. Krabill, J. G. Collins, L. E. Link, R. N. Swift and M. L. Butler, "Airborne laser topographic mapping results," *Photogrammetric Engineering and Remote Sensing*, vol. 50, no. 6, pp. 685-694, 1984.

[7] H. Arp, J. C. Griesbach and J. P. Burns, "Mapping in tropical forests: a new approach using the laser APR," *Photogrammetric Engineering and Remote Sensing*, vol. 48, no. 1, pp. 91-100, 1982.

[8] H. Schreier, J. Lougheed, J. R. Gibson and J. Russell, "Calibrating an airborne laser profiling system," *Photogrammetric Engineering and Remote Sensing*, vol. 50, no. 11, pp. 1591-1598, 1984.

[9] R. F. Nelson, W. B. Krabill and G. A. Maclean, "Determining forest canopy characteristics using airborne laser data," *Remote Sensing of Environment*, vol. 15, pp. 201-212, 1984.

[10] H. Schreier, J. Lougheed, C. Tucker and D. Leckie, "Automated measurements of terrain reflection and height variations using an airborne infrared laser system," *International Journal of Remote Sensing*, vol. 6, no. 1, pp. 101-113, 1985.

[11] G. A. Maclean and W. B. Krabill, "Gross-merchantable timber volume estima-

tion using an airborne LIDAR system," *Canadian Journal of Remote Sensing*, vol. 12, no. 1, pp. 7-18, 1986.

[12] D. G. Leckie, "Advances in remote sensing technologies for forest surveys and management," *Canadian Journal of Remote Sensing*, vol. 20, pp. 464-483, 1990.

[13] D. M. Jacobs, D. L. Evans and J. C. Ritchie, "Laser profiles and aerial video data for forest assessments," *Proceedings, 1993 ACSM/ASPRS. Remote Sensing, Looking to the Future with an Eye on the Past*, pp. 135-141, 1993.

[14] M. Nilsson, "Estimation of tree heights and stand volume using an airborne LIDAR system," *Remote Sensing of Environment*, vol. 56, pp. 1-7, 1996.

[15] J. C. Ritchie, "Airborne laser altimeter measurements of landscape topography," *Remote Sensing of Environment*, vol. 53, pp. 91-96, 1995.

[16] I. C. S. Han, "Airborne laser scanning for elevation mapping," *Geomatics Info Magazine*, vol. 10, pp. 57-60, 1996.

[17] D. Leckie, "Possible airborne sensor, processing and interpretation systems for major forestry applications," *Proceedings First International Airborne Remote Sensing Conference*, vol. 2, pp. 159-169, 1994.

[18] H. Samet, *Application of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*, Addison Wesley, Reading, MA, 1990.

[19] H. Samet, "Hierarchical Representations of Collections of Small Rectangles," *ACM Computing Surveys*, vol. 20, pp. 271-309, 1988.

[20] R. Franke and G. M. Nielson, "Scattered data interpolation and applications: A tutorial and survey," *Geometric Modelling: Methods and their Application*, H. Hagen and D. Roller, eds., Berlin: Springer-Verlag, pp. 131-160, 1991.

[21] D. Shepard, "A two-dimensional interpolation function for irregularly spaced data," *Proc ACM 23rd National Conference*, pp. 517-524, 1968.

[22] R. Franke and G. M. Nielson, "Smooth interpolation of large sets of scattered data," *International Journal for Numerical Methods in Engineering*, vol. 15, pp. 131-160, 1980.

[23] L. Schumaker, "Fitting surfaces to scattered data," *Approximation Theory II*, C. Chui, L. Schumaker and G. Lorentz, eds., Wiley, N.Y., pp. 203-268, 1976.

[24] T. A. Foley and G. M. Nielson, "Multivariate interpolation to scattered data using delta iteration," *Approximation Theory III*, E. Cheney, ed., Academic Press, N.Y., pp. 419-424, 1980.

[25] G. Deslauriers and S. Dubuc, "Symmetric iterative interpolation process," *Constructive Approximation*, vol. 5, pp. 49-68, 1989.

[26] S. Dubuc, "Interpolation through an iterative scheme," *Journal of Mathematical Analysis and Applications*, vol. 114, pp. 185-204, 1986.

[27] H. Choi and R. Baraniuk, "Interpolation and denoising of nonuniformly sampled data using wavelet-domain processing," *Proceedings - IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1645-1648, 1999.

[28] I. Daubechies, *Ten Lectures on Wavelets*, New York: SIAM, 1992.

[29] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley, MA: Wellesley-Cambridge Press, 1996.

[30] Y. Meyer, *Wavelets and Operators*, Cambridge University Press, 1992.

[31] P. Abry and P. Flandrin, "On the initialization of the discrete wavelet transform algorithm," *IEEE Signal Processing Letters*, vol. 1, no. 2, pp. 32-34, 1994.

[32] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Prentice Hall, New Jersey, 1995.

[33] C. L. Lawson, "Software for C^1 surface interpolation," *Mathematical Software III*, J. Rice, ed., Academic Press, N.Y., pp. 161-194, 1977.

[34] L. L. Schumaker, "Triangulation Methods," *Topics in Multivariate Approximation*, C. K. Chui, L. L. Schumaker and F. I. Utreras, eds., Academic Press, N.Y., pp. 219-232, 1987.

[35] R. Sibson, "Locally equiangular triangulations," *Computing Journal*, vol. 21, no. 3, pp. 243-245, 1978.

[36] S. W. Sloan, "A fast algorithm for constructing Delaunay triangulations in the plane," *Advances in Engineering Software*, vol. 9, no. 1, pp. 34-55, 1987.

[37] C. Huang and T. Shih, "Improvements on Sloan's algorithm for constructing Delaunay triangulations," *Computers & Geosciences*, vol. 24, no. 2, pp. 193-196, 1998.

[38] M. Zlámal, "On the finite element method," *Numerical Mathematics*, vol. 15, pp. 394-409, 1970.

[39] D. F. Watson, "Computing the n-dimensional Delaunay triangulation," *The Computer Journal*, vol. 24, no. 2, pp. 167-172, 1981.

[40] H. Akima, "A method of bivariate interpolation and smooth surface fitting for values given at irregularly distributed points," OT Report, U.S. Govt. Printing Office, Washington, D.C., 1976.

VITA

Surname: Sasmal ***Given Names:*** Debasish

Place of Birth: India ***Date of Birth:*** May. 21, 1976

Educational Institutions Attended

University of Victoria, Canada	1998 to 2000
Indian Institute of Technology, Kharagpur, India	1994 to 1998

Degrees Awarded

B.Tech.	Indian Institute of Technology, Kharagpur, India	1998

Honors and Awards

University of Victoria Research Assistantship	1998-2000

Conference Publications

1. D. Sasmal, W. S. Lu, D. Shpak and A. Antoniou "Interpolation Techniques for a LIDAR Profiler System", *IEEE PACRIM Conference*, August. 1999.