1-1-2013

Analysis and Processing of Irregularly Distributed Point Clouds

Kamala Hunt Diefenthaler

University of South Carolina - Columbia

Follow this and additional works at: http://scholarcommons.sc.edu/etd

Recommended Citation


This Open Access Dissertation is brought to you for free and open access by Scholar Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact SCHOLARC@mailbox.sc.edu.
Analysis and Processing of Irregularly Distributed Point Clouds

by

Kamala Hunt Diefenthaler

Bachelor of Science
Coastal Carolina University 2006

Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in
Mathematics
College of Arts and Sciences
University of South Carolina
2013
Accepted by:
Peter Binev, Major Professor
Hong Wang, Committee Member
Joshua Cooper, Committee Member
Robert Sharpley, Committee Member
Brian Habing, Committee Member
Lacy Ford, Vice Provost and Dean of Graduate Studies
DEDICATION

I dedicate this work to my God, who is the beginning of knowledge, and I’m thankful for His peace which passes all understanding. To my husband Jason, thank you for the consistent encouragement and believing all along that I was good enough. To my parents who, pulled all nighters with me and were available anytime I called, I will never forget your sacrificial display of love for me. Lastly, thanks to the many friends and family who partnered and prayed with me along this journey. Your support has meant the world to me.
ACKNOWLEDGMENTS

I want to thank my advisor Peter Binev for his vision, time, and dedication. Thanks to L. Scott Johnson for his attention to detail, moral and technical support. The Interdisciplinary Mathematics Institute provided the computers, office space, and programming frameworks that made the research in this dissertation possible. This work was partially supported by NSF grants DMS-0915104 and DMS-1222390, MURI/ARO grant W911NF-07-1-0185, and DEPSCoR ONR#N00014-07-10978.
ABSTRACT

We address critical issues arising in the practical implementation of processing real point cloud data that exhibits irregularities. We develop an adaptive algorithm based on Learning Theory for processing point clouds from a stationary sensor that standard algorithms have difficulty approximating. Moreover, we build the theory of distribution-dependent subdivision schemes targeted at representing curves and surfaces with gaps in the data. The algorithms analyze aggregate quantities of the point cloud over subdomains and predict these quantities at the finer level from the ones at the coarser level.
TABLE OF CONTENTS

Dedication ................................................................. iii

Acknowledgments ........................................................... iv

Abstract ............................................................................... v

List of Tables ................................................................. viii

List of Figures ................................................................. ix

Chapter 1 Preliminaries ..................................................... 5
    1.1 Point Clouds ............................................................... 5
    1.2 Barycentric Coordinates and Bernstein Polynomials .............. 8
    1.3 Triangulations and Newest Vertex Bisection ......................... 10
    1.4 Subdivision ................................................................. 21

Chapter 2 Assimilation and Processing of Point Clouds .......... 26
    2.1 Theoretical Framework .................................................. 26
    2.2 Spherical Learning Setting and Notation ............................ 29
    2.3 Surface Approximation .................................................. 32
    2.4 Progressive Transmission of an Approximation .................... 42
    2.5 Numerical Results ....................................................... 48
<table>
<thead>
<tr>
<th>Chapter 3</th>
<th>DISTRIBUTION-DEPENDENT SUBDIVISION SCHEMES</th>
<th>53</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Approximation of Q over Intervals</td>
<td>55</td>
</tr>
<tr>
<td>3.2</td>
<td>Approximation of the Value-Dependent Quantities</td>
<td>78</td>
</tr>
<tr>
<td>3.3</td>
<td>Approximation of Q over Triangles</td>
<td>82</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 4</th>
<th>CONCLUSION</th>
<th>119</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Future Work</td>
<td>120</td>
</tr>
</tbody>
</table>

| Bibliography | | 121 |
List of Tables

Table 2.1 Wavelet Results (JPEG 2000) ........................................... 50
Table 2.2 Line Average Prediction ....................................................... 51
Table 2.3 Total Average Prediction ...................................................... 52
Table 2.4 Max and Min Average Prediction ........................................... 52

Table 3.1 Boundary Transformations of $S$ in $ab$-plane to the $uv$-plane .... 66
Table 3.2 Additional Transformations in the $ab$-plane to the $uv$-plane ........ 69
Table 3.3 Type 1 Triangle Measures from the boundaries in the $bd$-plane to $uv$-plane ................................................................. 91
Table 3.4 Type 2 Triangle Measures from the boundaries in $cd$-plane to the $uv$-plane ................................................................. 93
Table 3.5 Type 3 Triangle Measures from $bc$-plane boundaries to the $uv$-plane 94
Table 3.6 The Parametric Equations and Inverses for the Triangle Measures 94
Table 3.7 Type 1 Complementary Measures from $bd$ boundaries to $uv$ ...... 99
Table 3.8 Type 2 Complementary Measures from $cd$ boundaries to $uv$ ...... 100
Table 3.9 Type 3 Complementary Measures from $bc$ boundaries to $uv$ ...... 102
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The domain $X$ with initial partition $P_0$.</td>
</tr>
<tr>
<td>1.2</td>
<td>The <em>reference triangle</em> with vertices $v_B = (0,1), v_L = (-1,0)$ and $v_R = (1,0)$.</td>
</tr>
<tr>
<td>1.3</td>
<td>Subdivision of the triangle $\Delta = (v_B, v_L, v_R)$ into its children $\Delta_0 = (v_<em>, v_B, v_L)$ and $\Delta_1 = (v_</em>, v_R, v_B)$ by <em>newest vertex bisection</em>.</td>
</tr>
<tr>
<td>1.4</td>
<td>Illustration of the Hausdorff distance between the sets $X$ and $Y$.</td>
</tr>
<tr>
<td>1.5</td>
<td>Mask for the addition of vertex to a face in the 4-8 Subdivision Scheme.</td>
</tr>
<tr>
<td>1.6</td>
<td>Masks for the regular case 4-8 Subdivision Scheme when updating a vertex.</td>
</tr>
<tr>
<td>2.1</td>
<td>A triangle where the points $F$ lie in a low dimensional space.</td>
</tr>
<tr>
<td>2.2</td>
<td>Two triangles $\Delta_1$ and $\Delta_2$ with shared vertex $v^*$. The vertex is assigned two values.</td>
</tr>
<tr>
<td>2.3</td>
<td>The vertex $v^*$ is created when the black diagonal edge $E$ is subdivided. The red edges result from the subdivision of $E$. The vertices $v'_1$ and $v'_2$ lie on $E$, and the vertices $v_1$ and $v_2$ are the vertices opposite $E$.</td>
</tr>
<tr>
<td>2.4</td>
<td>A SICK Commercial LiDAR scanner.</td>
</tr>
<tr>
<td>2.5</td>
<td>Slightly distorted view of the VT point cloud between spherical and Cartesian coordinates. (81,150 pts).</td>
</tr>
<tr>
<td>2.6</td>
<td>Surface generated from the algorithm. From left to right a hallway, three trees, and a building are distinguishable. (13,781 triangles).</td>
</tr>
</tbody>
</table>
Figure 3.6  The union of $\Lambda$ and $\Upsilon$. Notice that together they cover the convex hull of $g$. ................................. 67

Figure 3.7  The top left is the set $S$ in the $ab$-plane colored according to the transformation $M$ and the top right is colored according to the transformation $M^C$. The bottom plot is the image in the $uv$-plane of the four regions $\Lambda_1$ (blue), $\Lambda_2$ (pink), $\Upsilon_1$ (green), and $\Upsilon_2$ (yellow). ................................. 70

Figure 3.8  The overlay of $\Gamma$ from the polynomial weighted scheme and the regions from the subdomain scheme. The blending scheme combines the results from the two schemes into one scheme. . . . 76

Figure 3.9  The first image depicts the average $y$-values of the 6 initial intervals from which the actual quantities are calculated directly from the point cloud $D$. The other images are levels 1, 2 and 3 of subdivision where the quantities are calculated according to the polynomial scheme described in Section 3.1. These plots illustrate the issue that occurs when strictly using the polynomial scheme. Notice the large errors that occur around the gap. ................................. 77

Figure 3.10  Graphs of the initial averages for $y$, $x$, and $x^2$ calculated from the actual point cloud $D_1$. ................................. 81

Figure 3.11  Level 1 of subdivision: Plots of the averages for $y$, $x$, and $x^2$ calculated only from the quantities of the parent and parent’s neighbors. ................................. 81

Figure 3.12  Level 2 of subdivision: Plots of the averages for $y$, $x$, and $x^2$ calculated only from the quantities of the parent and parent’s neighbors. ................................. 81

Figure 3.13  Level 3 of subdivision: Plots of the averages for $y$, $x$, and $x^2$ calculated only from the quantities of the parent and parent’s neighbors. ................................. 81

Figure 3.14  Comparision of the average $y$-value for level 10 of subdivision (blue) to the actual point cloud $D_1$ (magenta). ................................. 82

Figure 3.15  Graphs of the initial averages for $y$, $x$, and $x^2$ of $D_2$. ................................. 83

Figure 3.16  Graphs of the averages for $y$, $x$, and $x^2$ of $D_2$ at level 1 of subdivision. 83

Figure 3.17  Graphs of the averages for $y$, $x$, and $x^2$ of $D_2$ at level 2 of subdivision. 83
Figure 3.18 Graphs of the averages for $y$, $x$, and $x^2$ of $D_2$ at level 3 of subdivision. 83

Figure 3.19 Graphs of the average $y$ for level 10 of subdivision (blue) and the actual point cloud $D_2$ (magenta). 84

Figure 3.20 The graph is plotted in the $uv$ plane, and the blue region is $\Delta_w$. The triangle $\Delta_w$ indicates the vectors $q$ for which a weighted measure corresponds. 88

Figure 3.21 Illustration of a triangle measure of Type 1. 89

Figure 3.22 Illustration of a triangle measure of Type 2. 89

Figure 3.23 Illustration of a triangle measure of Type 3. 89

Figure 3.24 The $uv$-plane colored according to the types. Type 1 is blue and the symmetric case is light blue. Type 2 is purple and the symmetric case is light purple. Type 3 is yellow. 94

Figure 3.25 Illustration of a complementary measure of Type 1. 98

Figure 3.26 Illustration of a complementary measure of Type 2. 98

Figure 3.27 Illustration of a complementary measure of Type 3. 98

Figure 3.28 The $uv$-plane colored according to the triangle and complementary measure types. $R_i$ are triangle measures and $T_i$ are complementary measures for $i = 1, 2, 3$. 103

Figure 3.29 The $uv$-plane highlighting the overlap of the weighted measures and the subdomain measures. 108

Figure 3.30 The point cloud $D_1$ and the triangulation $T^*$. Note that measure over each can be approximated well by the triangle and complementary representative measures discussed in Section 3.3. 112

Figure 3.31 The average values over the triangles in the triangulation $T^*$ and for the point cloud $D_1$. 112

Figure 3.32 The rows depict the subdivision levels 1 through 3 of $T^*$. The red plots are the average $y$-value. The green plots are the average $x_1$-values and the blue plots are the average $x_2$-values. 113

Figure 3.33 Subdivision level 10 (the limit surface). 114
Figure 3.34 The point cloud $D_1$ colored red according to the average $y$-value, colored green according to the average $x_1$-value, and then colored blue according to the average $x_2$-value. \hspace{1cm} 114

Figure 3.35 The point cloud $D_2$ and the triangulation $\mathcal{T}^*$. The better these initial measures over the triangles are represented by the triangle and complementary measures discussed in Section 3.3, the better the limiting surface of the algorithm. \hspace{1cm} 114

Figure 3.36 The average values over the triangles in the triangulation $\mathcal{T}^*$ and for the point cloud $D_2$. \hspace{1cm} 115

Figure 3.37 The rows depict the subdivision levels 1 through 3 of $\mathcal{T}^*$. The red plots are the average $y$-value. The green plots are the average $x_1$-values and the blue plots are the average $x_2$-values. \hspace{1cm} 115

Figure 3.38 Subdivision level 10 (the limit surface). \hspace{1cm} 116

Figure 3.39 The point cloud $D_2$ colored red according to the average $y$-value, colored green according to the average $x_1$-value, and then colored blue according to the average $x_2$-value. \hspace{1cm} 116

Figure 3.40 The image on the left depicts the triangle with Lebesgue measure over $A$. The image on the right depicts the subdomain representative measure found by the algorithm. This poor representation leads to visible errors as we refine. \hspace{1cm} 117

Figure 3.41 The first plot is the average $y$-values of the rectangular point cloud with $\mathcal{T}^*$. The plot in the middle is the algorithm result after 5 additional levels of subdivision. The third plot is after 15 levels of subdivision. Notice the error around the corners of the rectangle. \hspace{1cm} 118
INTRODUCTION

Point clouds are sets of unstructured points with no specific ordering or connections. In typical settings relevant to several applications, the data points correspond to a surface. For instance, terrain data has that the data points represent a location on earth. The data points are received by processing images, video, manual measurements, etc. One of the most promising ways of receiving terrain data is through LiDAR, which is a remote sensing technology that measures distance by illuminating a scene with a laser and analyzing the reflected light. Research problems often associated with this type of point clouds include: efficient feature extraction, surface reconstruction, and visualization. Several approaches such as subdivision, implicit surfaces, and variational surfaces have been developed toward these goals for point clouds with nearly uniform sampling at a rate much higher than the required resolution for reconstruction. When subjected to a relatively low sampling rate, LiDAR scanners commonly produce point clouds with undesirable properties such as missing data, occlusions, or uneven distribution. To address these issues, we introduce mathematical theory and tools for processing irregularly distributed data.

The content of this dissertation addresses two main problems. The first is the processing of point clouds that are regularly distributed in a topology different than the one provided by the standard Cartesian grid. The second problem is the approximation of distribution-dependent quantities during subdivision. A new paradigm is introduced for building subdivision schemes that take into account the distribution of the measure, thereby yielding schemes tailored to unevenly distributed data.

The first problem is focused on surface reconstruction and starts with learning
in a topology different from the Cartesian space. We assume the point clouds are sensed from a surface and are functional in the topology of the scanning process. In particular, we consider spherical learning, which is a natural approach for analyzing fixed or slow moving LiDAR scans. Fixed scans collect data in terms of two angles and the distance from the sensor which result in nearly regular distribution of points in spherical coordinates, yet the objects being sensed are mostly rectangular. In Cartesian coordinates, these point clouds are very irregular, as some regions are dense and others are sparse. Our algorithm partitions the surface in spherical space then builds piecewise linear approximations over the partition in Cartesian space. The first step is to transform the point cloud to the underlying topological space (in this case spherical) and generate a family of partitions using newest vertex bisection, which is organized in a multiresolution tree structure. The error is measured in the standard topology (in this case Cartesian). Thus we fit a Cartesian plane on each cell of the partition using the aggregate quantities of the cell. It is important to note that the basis of the algorithm calculations is the aggregate quantities of the point cloud data and not the individual data points. The aggregate quantities ought to characterize the local behavior of the data. We consider the moments of the point cloud within a subdomain as the aggregate quantities because the moments are useful tools in understanding the behavior of the points. Since we use the quantities and not the individual data points, we can easily process streaming data. When new points are added to the point cloud, we adjust the numerical values of the aggregate quantities and do not need to introduce a new object. Moreover, we develop a new method for encoding the surface approximation by introducing nonlinearity into the multiresolution tree itself.

Our algorithms are based on the theory and methodology in Learning Theory by Binev, Cohen, Dahmen, DeVore, and Temlyakov (see [4] and [2]). These papers focus on the approximation of the surface constructed over the partition cells. Their
work has provided the theoretical framework and tools from which we build our algorithm. Some explanation of the Learning Theory estimates in [4] and [2] as well as a description of the results will be given, but we will not repeat the estimates as they are very similar to what has been done and there is no added contribution for our particular properties. The estimates for piecewise constant approximations are optimal. The authors achieve similar results for piecewise polynomial approximations when a regularity condition is imposed on the distribution. Our practical algorithms use techniques such as singular value decomposition (SVD) to ensure that the process is stable.

The second problem of distribution-dependent subdivision schemes focuses on the issue of visualization. The film and video game industries, for example, often employ subdivision as a prominent technique for efficient visual representation of computerized objects. Subdivision surfaces connect the ideas of continuous and discrete models. Notable examples of subdivision schemes include: Catmull - Clark [7], Doo - Sabin [9], and Loop [12]. In Section 1.4 we discuss the one-dimensional example of subdivision based on splines and discuss the two-dimensional example of 4-8 Subdivision by Velho and Zorin [15]. Most of this work concerns subdivision schemes for dense and evenly distributed data and is driven by the functional values. They have a very different flavor than the methods we discuss. Our approach in Chapter 3 is to approximate not only the surface but also the distribution-dependent quantities, which we define to be the desired quantities that depend only on the location and distribution of the points in the data set. As in the first problem, the moments are the local aggregate quantities and drive the development of the subdivision schemes. We draw analogies from polynomial reproduction to reproduction of the aggregate quantities of particular types of measures. This method of learning the distribution-dependent quantities is a novel method, so we start by building the foundation. This dissertation presents algorithms and results for curve subdivision over intervals and
The dissertation is organized in the following manner. In Chapter 1, we supply the definitions and background information used in the remaining chapters. In Chapter 2, we tailor the general algorithms in Learning Theory [4] and [2] to the case of spherical underlying topology. A nonlinear procedure to encode our surface approximation is described. Chapter 3 explores the idea of learning the measure to adjust for gaps and unevenly distributed point clouds. The methods explained in Chapter 3 serve as a launchpad for future work. Finally, Chapter 4 summarizes our results and outlines some of the many possible directions for future study.

This work was partially supported by NSF grants DMS-0915104 and DMS-1222390, MURI/ARO grant W911NF-07-1-0185, and DEPSCoR ONR#N00014-07-10978.
Chapter 1

Preliminaries

1.1 Point Clouds

A point cloud $D$ is a finite set of $d + 1$ dimensional data points. Each point $p \in D$ is of the form $p = (x, y)$ where $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$. We assume that there are bounded regions $X \subseteq \mathbb{R}^d$ and $Y \subseteq \mathbb{R}$ such that for all possible point clouds $D$ and points $p \in D$ we have that $x \in X$ and $y \in Y$. For convenience we assume $X$ is a $d$-dimensional cube, as for every bounded region there is a hypercube containing it. We purposefully do not consider a point $p$ to be a single vector in $\mathbb{R}^{d+1}$ because we assume that $p$ is the result of probing a real valued functional surface, namely $s$, at $x$. Sometimes we will refer to $y$ as the value at $x$, and at times it is representative of the height of a terrain at the location $x$. In the case of perfect acquisition, we would have $y = s(x)$, but in reality we expect to have errors in acquiring both the value $y$ and its position $x$.

Mathematically, the point cloud is much more than a simple set of points. Here we establish the mathematical framework for the distribution of the points and the relation between the point cloud and the surface. We assume that $D$ consists of independent random observations that are identically distributed according to a probability measure $\rho$ on $X \times Y \subseteq \mathbb{R}^{d+1}$. Probability measures are the same as the more general notion of positive measure with the added condition that the measure of the entire probability space must be equal to one. The marginal probability measure on
X is denoted by \( \rho_X \) and is defined for \( S \subseteq X \) as

\[
\rho_X(S) := \rho(S \times Y).
\]

When extending \( X \) to a \( d \)-dimensional cube, we also extend \( \rho_X \) with zeros outside of the original set. The measure \( \rho_X \) is considered unknown. In Chapter 2 we assume that the underlying topology of \( \rho_X \) is spherical, and in Chapter 3 we approximate \( \rho_X \) on subdomains each time we subdivide. We let \( L_2(X, \rho_X) \) denote the space that consists of all functions from \( X \) to \( Y \) which are square integrable with respect to \( \rho_X \). We also have that \( d\rho(x, y) = d\rho(y|x)d\rho_X(x) \), where \( \rho(y|x) \) is the conditional probability measure on \( Y \) given \( x \in X \). The regression function is the conditional expectation of \( y \) at \( x \)

\[
s_\rho(x) := \int_Y y d\rho(y|x).
\]

In this dissertation, we assume that there is an \( M \) such that \( |y| \leq M \) almost surely, which implies that \( |s_\rho| \leq M \) almost everywhere with respect to \( \rho_X \). Furthermore, \( s_\rho \) is the function in \( L_2(X, \rho_X) \) that minimizes the risk functional

\[
\int_{X \times Y} |y - f(x)|^2 d\rho = \|y - f(x)\|_{L_2(X, \rho_X)}^2
\]

(see, [3]). In this model we consider \( \rho \) to be a measure concentrated around the surface of interest, \( s \), and will assume that \( s_\rho \) is the surface that represents the point cloud. Ideally there is no measurement error in the acquisition of the point cloud, so \( s_\rho = s \). Therefore, we disregard the model error \( \|s_\rho - s\|_{L_2(X, \rho_X)} \) in our future considerations and our objective is to approximate \( s_\rho \). We denote our approximation of \( s_\rho \) by \( \tilde{s} \).

The points in \( D \) will be partitioned into subdomains. Throughout the paper we calculate important aggregate values from the group of points in each given subdomain. The main idea is that on a subdomain \( R \), we only examine these aggregate quantities representing the points in \( R \) and ignore the individual points. Let \( Q(R) \) be
the aggregate representation of the points in the subdomain $R$. These aggregate quantities should be quantitative measures of the shape of the points in the subdomain. The moments of $\rho_X$ are a natural choice to include in our aggregate representation. When $d = 1$ and $x \in \mathbb{R}$ the moments are:

$$M_i(R) = \int_R x^i \ d\rho_X \approx \sum_{x \in R} x^i \quad \text{for } i = 0, 1, 2, 3, \ldots \quad (1.1)$$

We will also consider the moment $M_y(R) = \int_R y \ d\rho_X$. From these moments several useful quantities can be calculated, namely the average values and variance. The zero moment gives us the size of the measure $\rho_X$ on $R$. $M_1/M_0$ gives us the average value of $x$ on $R$, and $M_y/M_0$ is the average value of $y$ on $R$. The variance of $x$ on $R$ is $M_2 - \left(\frac{M_1}{M_0}\right)^2$.

The moments for $x = (x_1, x_2) \in \mathbb{R}^2$ are:

$$M_{ij}(R) = \int_R x_1^i x_2^j \ d\rho_X \approx \sum_{x \in R} x_1^i x_2^j \quad \text{for } i = 0, 1, 2, 3, \ldots \quad (1.2)$$

Expanding on the values discussed for $\mathbb{R}$, we have the covariance which is $M_{11} - M_{10}M_{01}$. With these quantities we can build the covariance matrix. The smallest eigenvalue of the covariance matrix is another valuable quantity to be used in some algorithms.

A quantity $q$ is additive if $q(R_1) + q(R_2) = q(R_1 \cup R_2)$ for disjoint subdomains $R_1$ and $R_2$. We desire that all of the quantities in $Q$ are additive. If a quantity in $Q$ is not additive, then future calculations will require more computations. For additive quantities, knowing the quantities over finer subdomains allows for quick calculation of the coarser subdomain quantities. For our purposes, $Q(R)$ consists of the moments, and from the moments we calculate the other quantities as necessary. We choose the moments because they are additive and descriptive of the points they represent. We do not let the $Q(R)$ be the averages, variance, etc. because these quantities are not additive.
Point Clouds from LiDAR

The point clouds considered in Chapter 2 are LiDAR scans, where the sensor remained in the same or nearly the same position. These point clouds generally are irregularly distributed in Cartesian space. For instance, the regions where objects are sensed are relatively dense and the regions without a sensed object are relatively sparse. In spherical space, these types of point clouds generally are evenly distributed with the exception of gaps or holes in the data.

Two common issues from real world LiDAR scans cause gaps in the point cloud. The first is occlusion, which occurs when one object blocks the sensor from acquiring points in a region. The point cloud will only contain data along the line of sight of the sensors. In urban areas, it is common to have only points over one or two sides of a building, the other sides being occluded. The second issue is no returns. Sometimes, no information is transmitted back to the sensor, and thus no return is recorded. Where no returns occur, holes will appear in the point cloud. One potential cause of no returns is water. Infrared beams used by LiDAR tend to be absorbed by water and show very weak or no return signal when they hit rain, snow, thick smoke or very thick haze.

1.2 Barycentric Coordinates and Bernstein Polynomials

A simplex is the convex hull of \( d + 1 \) vertices \( \mathbf{v}_i \in \mathbb{R}^d, i = 0, \ldots, n \), and is denoted by \( \Delta = (\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_d) \), a \( (d + 1) \)-tuple of \( d \)-dimensional vectors. If \( \{\mathbf{v}_i - \mathbf{v}_0\}_{i=1}^d \) are linearly dependent, then the simplex is degenerate. We want to emphasize that points will only refer to elements of the point cloud and never to vertices. Next we give definitions and some properties of barycentric coordinates and Bernstein polynomials. More information and basic proofs can be found in Farin [11].

\textbf{Definition 1.1.} Given a simplex \( \Delta = (\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_d) \) the \textit{barycentric coordinates} of a
point \( p \in \mathbb{R}^d \) are \((\lambda_0, \lambda_1, \ldots, \lambda_d)\) such that \( p = \lambda_0 v_0 + \lambda_1 v_1 + \cdots + \lambda_d v_d \) and \( \sum_{i=0}^{d} \lambda_i = 1 \).

Although an abuse of notation, we write \( \lambda_v(\Delta, p) \) as the barycentric coordinate of \( p \) in the simplex \( \Delta \) with respect to the vertex \( v \).

The usefulness of the barycentric coordinates is based on their simplicity and flexibility in any \( d \). Each barycentric coordinate of \( p \) is essentially the weight of the corresponding vertex for \( p \). Furthermore, the barycentric coordinates with respect to a nondegenerate simplex are unique for each point because we enforce that \( \sum_{i=0}^{d} \lambda_i = 1 \) in the definition. When \( p \) is near a vertex we have that the corresponding barycentric coordinate will be large, and \( p \) far away from a vertex will have a small barycentric coordinate. In fact, the barycentric coordinates of the vertices of a non-degenerate simplex are \( \lambda_{v_0}(\Delta, v_0) = (1, 0, \ldots, 0), \lambda_{v_1}(\Delta, v_1) = (0, 1, 0, \ldots, 0), \ldots, \lambda_{v_d}(\Delta, v_d) = (0, 0, \ldots, 1) \). If all of the barycentric coordinates are positive, then \( p \) lies inside the convex hull of the vertices. On the other hand, \( p \) lies outside when at least one coordinate is negative. Furthermore, barycentric coordinates are affine invariant. That is to say that they do not change when the simplex and point undergo the same affine transformation.

The barycentric coordinates \( \lambda_1 \) to \( \lambda_d \) are found by solving the following system:

\[
\begin{pmatrix}
  v_1 - v_0 & v_2 - v_0 & \cdots & v_d - v_0
\end{pmatrix}
\begin{pmatrix}
  \lambda_1 \\
  \lambda_2 \\
  \vdots \\
  \lambda_d
\end{pmatrix}
= p.
\]

The matrix is invertible since the \( v_i \) are the vertices of a non-degenerate simplex. Then the \( \lambda_0 \) is found by \( \lambda_0 = 1 - \sum_{i=1}^{d} \lambda_i \). For a triangle \((v_0, v_1, v_2)\), the formulas are simply

\[
\lambda_0 = \frac{\text{area}(p, v_1, v_2)}{\text{area}(v_0, v_1, v_2)}, \quad \lambda_1 = \frac{\text{area}(v_0, p, v_2)}{\text{area}(v_0, v_1, v_2)}, \quad \lambda_2 = \frac{\text{area}(v_0, v_1, p)}{\text{area}(v_0, v_1, v_2)}.
\]
Barycentric coordinates are used to define the Bernstein polynomials, which are utilized in Chapter 3. The most important property of the degree \( n \) Bernstein polynomials is that they form a basis for all the polynomials of degree \( n \).

**Definition 1.2.** Let \( (\lambda_0, \lambda_1, \ldots, \lambda_d) \) be the barycentric coordinates of \( p \). The Bernstein polynomials of degree \( n \) over a simplex of degree \( d \) are defined by

\[
B^n_t(p) = \binom{n}{t} \lambda_0^{t_0} \lambda_1^{t_1} \cdots \lambda_d^{t_d}
\]

where \( t = (t_0, \ldots, t_d) \in \mathbb{N}^{d+1}, \sum_{i=0}^{d} t_i = n \), and \( \binom{n}{t} = \frac{n!}{t_0! t_1! \cdots t_d!} \).

**Example 1.3.** The Bernstein polynomials of degree \( n \) over the interval \([a, b]\) are:

\[
B^n_t(x) = \binom{n}{i} \frac{(x - a)^i (b - x)^{n-i}}{(b - a)^n}.
\]

Some properties of the barycentric coordinates directly carry over to the Bernstein polynomials:

- **affine invariance**

- \( \sum_t B^n_t(p) = (\lambda_0 + \cdots + \lambda_d)^n = 1 \)

- \( p \) in the simplex if and only if \( B^n_t(p) \) is positive for all \( t \).

All of the Bernstein polynomials are symmetric with respect to the center of the simplex.

1.3 **Triangulations and Newest Vertex Bisection**

This section explains the method we use to partition the domain \( X \) containing the point cloud \( D \) in \( \mathbb{R}^2 \). The first topic is to generate partitions using the *newest vertex bisection* method of triangle subdivision. We discuss the natural master tree that corresponds to the partitions generated by the *newest vertex bisection*. Together, the tree structure and the subdivision method allow for multiresolution analysis of
the subdivision surfaces. Then the topic shifts to adaptively choosing a particular partition from the *newest vertex bisection* method based upon the aggregate quantities of the point cloud. The last topic discusses the usefulness of *conformal* partitions and the procedure to ensure that the partition is *conformal*.

**Definition 1.4.** A partition $P$ of $X \subseteq \mathbb{R}^d$ is *conformal* if any two simplices in $P$ are either disjoint or intersect at a simplex of lower dimension.

**Newest Vertex Bisection**

*Newest vertex bisection* has been used successfully in adaptive finite element methods (AFEM) to establish convergence and rates of convergence (see, Morin [13] and [5]). Furthermore, *newest vertex bisection* has generalizations to $\mathbb{R}^3$ by tetrahedral bisection [1] and to $\mathbb{R}^d$ by simplex bisection [14]. The tree structure benefits carry to $\mathbb{R}^d$ in [6].

For our convenience we assume $X$ is a square, or in $\mathbb{R}^d$ a $d$-dimensional cube. The *newest vertex bisection* method subdivides a triangle by bisecting the edge opposite the newest vertex. So to start, we need an initial partition $P_0$ and an assignment of newest vertices. To keep the partition simple, we define the initial partition of the square $X$ to be two right-isosceles triangles as seen in the figure.
Figure 1.2  The *reference triangle* with vertices $v_B = (0, 1), v_L = (-1, 0)$ and $v_R = (1, 0)$.

Figure 1.3  Subdivision of the triangle $\Delta = (v_B, v_L, v_R)$ into its children $\Delta_0 = (v_s, v_B, v_L)$ and $\Delta_1 = (v_s, v_R, v_B)$ by *newest vertex bisection*.

We label each vertex corresponding to a right angle as the newest vertex in its respective triangle. *Newest vertex bisection* with these initial conditions gives us that all triangles in the partitions will be right-isosceles. Furthermore, the newest vertex of a triangle will always be at the right angle.

Let \( \Delta \) be a right-isosceles triangle. Then \( \Delta \) is uniquely identified by its vertices \((v_B, v_L, v_R)\), where \( v_B \) is the newest vertex. Any right-isosceles triangle can be transformed to the *reference triangle* with vertices at \((0, 1), (-1, 0), \) and \((1, 0)\). The vertex corresponding to \((0, 1)\) is labelled \( v_B \), \((-1, 0)\) is labeled \( v_L \), and \((1, 0)\) is labeled \( v_R \).

We subdivide the parent triangle \( \Delta = (v_B, v_L, v_R) \) by bisecting the edge opposite the newest vertex and adding a new vertex \( v_* \) at the midpoint. The left child is \( \Delta_0 = (v_*, v_B, v_L) \) and the right child is \( \Delta_1 = (v_*, v_R, v_B) \). The vertex \( v_* \) is the newest vertex of the children, as the name would suggest. Bisecting the hypotenuse
of a right-isosceles triangle always results in a right angle and the finer triangles created are right-isosceles. Therefore the newest vertex is always at the right angle of a triangle and all of the triangles resulting from further subdivision will also be right-isosceles. If a different initial partition or labeling was chosen, then we may not have that all of the triangles are right-isosceles. The framework shown here will be relevant for more general partitions when the initial setup is different.

Now we will describe the tree structure that accompanies the newest vertex bisection. The initial partition $P_0$ consists of two triangles $\Delta_0$ and $\Delta_1$. In fact, we write $P_0 = \{\Delta_0, \Delta_1\}$. Construction of our master tree $T$ begins with an empty root $\Delta_{\emptyset}$. Then the left child of the root is $\Delta_0$ and the right child is $\Delta_1$ which are the triangles that make up $P_0$. Next, $\Delta_0$ is subdivided into its children $\Delta_{00}$ and $\Delta_{01}$ by the method described above. The master tree is expanded by appending $\Delta_{00}$ and $\Delta_{01}$ to $\Delta_0$ in $T$. Continuing to subdivide subsequent generations by newest vertex bisection appends more and more children nodes to $T$. Thus, $T$ is an infinite rooted tree that is strictly binary. The set of nodes of $T$ are all of the triangles which can be obtained by a sequence of subdivisions starting from $P_0$. Once we have $P_0$ and the initial newest vertex assignment, then the master tree $T$ is predetermined.

Full binary trees are defined to be binary trees in which all nodes have either two children or no children. We define a pruned subtree to be a subtree that contains the root. Let $S$ be the set of all finite, full, and pruned subtrees of $T$, and let $\mathcal{P}$ be the set of all partitions of $X$ generated by the newest vertex bisection. For any tree $S$ let $\mathcal{N}(S)$ be the set of the nodes of $S$ and $\mathcal{L}(S)$ denote the leaves of $S$.

**Definition 1.5.** $L \subseteq \mathcal{N}(T)$ is a partition in $\mathcal{P}$ if and only if $\bigcup_{\Delta_i \in L} \Delta_i = X$ and the intersection of any two distinct triangles in $L$ has planar Lebesgue measure zero.

**Lemma 1.6.** Define $R : S \to \mathcal{P}$ by $R(S) = P$ whenever $\mathcal{L}(S) = P$. Then $R$ is bijective.
Proof. Let $P$ be any partition in $\mathcal{P}$. Then by definition $P \subseteq \mathcal{N}(T)$. Create $S$, which will be the pre-image of $P$, by appending to $P$ all of the ancestors of the triangles in $P$, so that $S$ is a pruned subtree of $T$. If any triangle in $P$ were the parent of another triangle in $P$, then $P$ would not be a partition. So $P = \mathcal{L}(S)$. It remains to show that $S$ is full. Suppose $S$ were not full. Then there exists a node $\Delta \in S$ with children $\Delta_0$ and $\Delta_1$ in $T$ such that only one child is in $S$. Without loss of generality, assume $\Delta_0$ is in $S$. Then $\Delta, \Delta_1$, and the children of $\Delta_1$ are not in $P$. This implies that $P$ does not cover all of $X$. Therefore $S \in S$ which implies $R(S) = P$. Hence $R$ is surjective. Let $S_1$ and $S_2$ be in $S$ such that $R(S_1) = R(S_2)$. Then $\mathcal{L}(S_1) = \mathcal{L}(S_2)$. Since they are full binary trees with the same root and same leaves, we have that $S_1 = S_2$. So $R$ is bijective.

Now that the one-to-one correspondence between $S$ and $\mathcal{P}$ has been established, we introduce two operations that will allow us to describe families of partitions: splitting and merging. Let $\Delta$ be a triangle with children $\Delta_1$ and $\Delta_2$. For $S \in S$ and $\Delta_1, \Delta_2 \in \mathcal{L}(S)$, the operation $\text{merge} (\Delta_1, \Delta_2)$ removes both $\Delta_1$ and $\Delta_2$ from $S$, which implies that $\Delta$ is now in $\mathcal{L}(S)$. For $S \in S$ and $\Delta \in \mathcal{L}(S)$, the operation $\text{split}$ (or $\text{subdivide}$) $\Delta$, does the opposite in that it appends the children of $\Delta$ in the master tree to $S$. Merging two triangles of a partition decreases the number of triangles in the partition by one, while splitting a triangle increases the number of triangles by one. In fact, any partition in $\mathcal{P}$ can be obtained from any other partition in $\mathcal{P}$ by a sequence of splits and merges.

The structure of the master tree and the subdivision method allow for multiresolution analysis, since the partitions of finer levels maintain the vertex locations of the coarser levels. Subdivision by newest vertex bisection starts with a base mesh of two triangles and recursively obtains finer meshes, not necessarily uniformly. Finer meshes correlate to increased depth in the master tree. Each time a triangle is subdivided, the location of the vertices in the parent remains fixed and a new vertex is
added, which implies that a fine mesh contains all of the coarser meshes. A surface can be described at varying depths of the master tree, specifically coarse approximation to fine. Applications of multiresolution meshes include progressive transmission and level of detail control. Progressive transmission begins with a low resolution approximation with very few details, and then progressively improves by adding more details. Level of detail is controlled by the addition or removal of subdivision branches.

Of primary interest to any implementation of this newest vertex bisection method is having an efficient way to calculate and maintain the quantities we store in each node of the tree. The simplest way is also the most computationally expensive: when a triangle is split, sort the points from the parent into each child and then calculate the quantities for each child directly from those points. A more efficient algorithm in which a tree skeleton is constructed and the quantities are calculated directly on the finest level and then are added together for coarser levels is described in [6]. Additivity of the quantities is a necessary property in order to use this more efficient algorithm. The algorithm in Chapter 3 needs neither a tree skeleton nor continual sorting of the points. We calculate all the quantities of the finer level directly from the coarser level.

**Adaptive Partitions**

Using the information from the point cloud, we adaptively choose a partition $P_A \in \mathcal{P}$ that satisfies the aim of the application. That is: we tailor the partition to both the specific requirements of the application and the information present in the point cloud. For instance, the partition should be subdivided more (making a finer mesh) near areas of rapid change or important details and subdivided less (remaining coarser) over areas of slight change. The criteria driving the choice of $P_A$ are motivated by the factors that will produce a quality approximation of $s_\rho$. There are two main approaches to arrive at an adaptive partition: Coarse-to-Fine and Fine-to-Coarse. The Coarse-
The Coarse-to-Fine approach is driven by the subdivision criteria discussed in Section 2.3 while the Fine-to-Coarse approach is driven by the merging criteria. The tree structure allows for a combination of the two approaches using both splitting and merging. The adaptive partition $P_A$ is the partition resulting from repeatedly subdividing $P_0$ until none of the leaf nodes meet the subdivision criteria, or similarly for Fine-to-Coarse: repeatedly merging a finest-level-of-detail partition until none of the leaf nodes meet the merging criteria.

In the Coarse-to-Fine approach, we start with the initial partition $P_0$ and, based on subdivision criteria derived from the quantities of the triangles, we mark triangles for subdivision. Then all marked triangles are subdivided. When conformal partitions are desired, we also split neighboring triangles as needed to resolve hanging vertices, even when those triangles do not meet the subdivision criteria themselves. See Conformal Partitions in the following section. After each stage of subdivision, the quantities are recalculated. Triangles are marked and subdivided repeatedly until all of the triangles in the partition satisfy the subdivision criteria. One benefit of the Coarse-to-Fine approach is that only the necessary triangles in $T$ need to be calculated. For very large point clouds, this saves a significant amount of computation.

In the Fine-to-Coarse approach, we start with some large finite subtree of the newest vertex bisection tree $S \subset T$, in which $S$ is large enough that no finer details are necessary. Then the quantities are calculated for the leaves of $S$. Triangles are then merged based on the merging criteria. If conformal partitions are desired, then only diamonds and boundary triangles are merged, and forced mergings may occur. The Fine-to-Coarse approach allows for easy encoding and computation of the quantities.

The criteria used to mark a triangle for subdivision or merging vary and can be tailored to the specific goals of the application. Most commonly an error term is defined for a triangle. The error could be the variance, the sum of the residuals,
and so on. Whatever criteria is used, it is important that the quantities involved are easily computed.

**Conformal Partitions**

Applications in which the sensed surface is supposed to be continuous desire that the surface approximation is also continuous. The partitions generated so far might not produce a continuous approximation. More specifically if a vertex of one triangle intersects the edge of another triangle, then a gap may occur between the vertex and the edge. Such a vertex $v$ is called a hanging vertex of $P$, as $v$ appears in the interior of an edge in $P$. In conformal partitions (Def. 1.4) hanging vertices are not permitted, which guarantees that the approximations built on conformal partitions are continuous. In $\mathbb{R}^2$, triangles of a conformal partition $P$ are permitted to intersect only at a vertex or an edge. This section discusses two procedures of receiving conformal partitions: completion after subdivision and force-splitting. If the partitions are not required to be conformal then it is not necessary to implement either procedure. In Chapter 2, we apply the force-splitting procedure to produce a conformal partition. However, our sensed surface is composed of different objects (e.g., tree, building) meaning only parts of the surface would be continuous. So we adjust for discontinuities using criteria discussed in Section 2.1, and our final partition will not be conformal.

Completion after subdivision is a procedure to turn a non-conformal partition into a conformal one. This is a typical procedure used in adaptive finite element methods that mark triangles for subdivision and then split all marked triangles. After all of the subdivisions are made to satisfy the subdivision criteria, the method finishes with a completion step. In the completion step, the partition is further subdivided until there are no hanging vertices. The number of triangles subdivided in the completion step is shown to be bounded by a constant multiple of the number of triangles in the
uncompleted partition (see, [5]).

The second method consistently maintains the conformality of the partitions using force-splits (see, [10]). Thus force-splitting is particularly useful for the Coarse-to-Fine approach to adaptive partitions. In order to prevent hanging vertices along the way, only certain triangles are eligible for immediate subdivision, and when a triangle is not eligible a neighbor will be forced to subdivide as well. A benefit is that the partition is conformal after each subdivision. For the convenience of the reader, we include some details here.

We remain in the same setting as in Section 1.3 using newest vertex bisection. Recall that the cells are right-isosceles triangles and are marked for subdivision based on the subdivision criteria in Section 2.3. Now we force additional triangles to split. Although we desire to subdivide a triangle $\Delta$, we may have to subdivide other triangles first so that hanging vertices are avoided. A triangle $\Delta$ is eligible for subdivision under one of the two following circumstances. First, $\Delta$ is a boundary triangle, which is defined as a triangle whose hypotenuse is on the boundary of $X$. Second, $\Delta$ shares a hypotenuse with another triangle from the same level in $T$. This case is referred to as a diamond, consisting of $\Delta$ and the triangle sharing the same hypotenuse. In the case of a diamond, the subdivision of $\Delta$ forces the other triangle to immediately split.

The idea is that in order to subdivide a non-eligible triangle $\Delta^*$, we first must subdivide other eligible triangles until $\Delta^*$ becomes eligible. Let $\Delta^*$ be a non-eligible triangle marked for subdivision. Then we define the following recursive procedure terminating when $\Delta^*$ becomes eligible and the diamond is split. Since $\Delta^*$ is not eligible we have that $\Delta^*$ has a neighbor $\Delta'^*$ from a coarser level. Geometrically, the depth of $\Delta^*$ is required to be exactly one more than the depth of $\Delta'^*$. If $\Delta'^*$ is eligible for subdivision, then we subdivide $\Delta'^*$ making $\Delta^*$ eligible for subdivision. It is possible that $\Delta'^*$ is not eligible for subdivision. In this case we mark $\Delta'^*$ for subdivision, and
recursively repeat the process above marking until we arrive at a triangle that is both marked and eligible. The triangles encountered during this recursion continue to decrease in depth, so the recursion always terminates.

**Singular Value Decomposition**

A singular value decomposition (SVD) provides a convenient way for breaking a matrix into a product of simpler matrices. The SVD factors any \( m \times n \) matrix \( A \) into the form \( U \Sigma V^T \) where \( U \) and \( V \) are unitary and \( \Sigma \) is diagonal. The diagonal entries of \( \Sigma \) are called singular values and are often denoted by \( \sigma_i \) for \( i = 1, \ldots, n \).

To visually understand SVD, consider the geometry in \( \mathbb{R}^2 \) for \( A \in \mathbb{R}^{2 \times 2} \) a full rank matrix. The image of the unit circle under multiplication by \( A \) is an ellipse. Then the singular values of \( A \) are the lengths of the semi-principal axes of the ellipse. The matrices \( U \) and \( V \) can be thought of as rotation matrices. It is possible for some singular values to be zero or near-zero. In our geometric perspective, a near-zero singular value means that the minor axis of the ellipse is so small that the ellipse is flattened very close to a line. Thus, in general dimensions, having a zero singular value implies that the data in \( A \) lies in a space with dimension less than \( n \).

For our purposes, the matrices with singular values near or equal to zero are of particular interest. In these circumstances one can consider the truncated SVD. Let \( t \) be a given threshold for the singular values, and \( A \in \mathbb{R}^{m \times n} \). Then suppose \( A \) has \( k \) singular values greater than \( t \). Then only the \( k \) column vectors of \( U \) and the \( k \) row vectors of \( V^T \) corresponding to the \( k \) largest singular values of \( \Sigma \) are calculated. The rest of the matrix is discarded. Thus \( A \approx \tilde{A} = U_k \Sigma_k V_k^T \). The approximate matrix \( \tilde{A} \) is the closest approximation to \( A \) that can be achieved by a matrix of rank \( k \). Naturally, if \( k = \text{rank}(A) \), then \( \tilde{A} = A \).

One application of SVD is solving a system of equations. Assume \( A \mathbf{x} = \mathbf{b} \) and we are solving for \( \mathbf{x} \). Then the SVD of \( A \) gives us that there exists a diagonal matrix \( \Sigma \)
and two unitary matrices $U$ and $V$ such that $A = U\Sigma V^T$. (Side note: When $A$ is real symmetric, we have that $U = V = U^T$ and that the entries of $\Sigma$ are nonnegative.)

$$Ax = b$$

$$U\Sigma V^Tx = b$$

$$\Sigma(V^Tx) = U^Tb.$$ 

Assuming the singular values of $A$ are nonzero, solving for $V^Tx$ is simple because $\Sigma$ is diagonal, and let $S$ denote this solution. Using the truncated SVD would also avoid any issues with solving for $V_i^Tp$. Then

$$V^Tx = S \quad \Rightarrow \quad x = VS.$$

There are several algorithms for calculating the SVD with stable performance for moderate dimensional matrices. We will be using the SVD only in three dimensions, so there is no problem using the SVD standard in MATLAB or other computation software.

**Hausdorff Distance**

The Hausdorff distance measures how far two subsets of a metric space are from each other, and is most often used when comparing two representations of the same three dimensional object. We use the Hausdorff distance when comparing our point cloud approximation to the original surface.

**Definition 1.7.** Let $X$ and $Y$ be two non-empty subsets of a metric space $(M,d)$. We define their Hausdorff distance $d_H(X,Y) = \max \left\{ \sup_{x \in X} \inf_{y \in Y} d(x,y), \sup_{y \in Y} \inf_{x \in X} d(x,y) \right\}$. 

For our purposes $M$ will be $\mathbb{R}^3$ and $d$ is the Euclidean distance. Simply put, the Hausdorff distance is the maximum of the distance of the farthest point in $X$ to $Y$ and the distance of the farthest point in $Y$ to $X$. Note that the Hausdorff distance is not generated by a norm.
Figure 1.4 Illustration of the Hausdorff distance between the sets $X$ and $Y$.

Often the $L^2, L^1$, or $L^\infty$ errors are used, but we argue that the Hausdorff distance offers a better measurement of the similarities between two surfaces in terrain approximation modeling. Take for instance two scenes that differ only by a telephone pole. A telephone pole is used in the illustration because the base of the pole is very small, and the height is large. The $L^2$ and $L^1$ errors would be small, yet the Hausdorff would be large. The $L^\infty$ error would also be large; however, a slight horizontal shift in the approximation could result in very large $L^\infty$ error. Therefore, we use the Hausdorff because differences like telephone poles are significant in our applications.

The presence of outliers greatly impacts the Hausdorff distance. Outliers may be present in the point clouds we analyze, but we use the Hausdorff metric to compare our surface approximation to the actual surface. In this case our approximation has already decreased the influence of the outliers, so the comparison of the surfaces are fairly resistant outliers.

1.4 Subdivision

Subdivision surfaces are becoming the standard surface models, and are used in a wide range of applications. Subdivision is easy to implement and computationally efficient. In this section we discuss splines [8] as a one dimensional example of subdivision and 4-8 subdivision [15] as a two dimensional example. Then we explore the idea of
polynomial reproduction and the influence it has on the development of subdivision schemes.

**Splines**

Splines are smooth piecewise polynomial curves of some designated degree with high continuity at the junctions. We show here how to define splines and how they can be generated through subdivision. We start at the very beginning with B-splines. B-splines are defined through repeated convolution,

\[ B^1(x) = \begin{cases} 
1 & 0 \leq x < 1 \\
0 & \text{otherwise}
\end{cases} \quad (1.4) \]

and then

\[ B^m(x) = B^1(x) \otimes B^{m-1}(x) = \int B^1(t)B^{m-1}(x-t)dt. \quad (1.5) \]

From this definition there are some important properties that follow. The first is that the B-spline of degree \( n \) is \( C^{n-1} \) continuous, and is a direct consequence of convolution with \( B^1 \). The second property is that B-splines obey a refinement equation. That is, a B-spline can be written as a linear combination of dilations and translations of itself. The refinement equation is given by

\[ B^m(x) = \frac{1}{2^{m-1}} \sum_{k=0}^{m} \binom{m}{k} B^m(2x - k). \quad (1.6) \]

For instance

\[ B^2 = \frac{1}{2} \left( \binom{2}{0} B^2(2x) + \binom{2}{1} B^2(2x-1) + \binom{2}{2} B^2(2x-2) \right) \]
\[ = \frac{1}{2} \left( B^2(2x) + 2B^2(2x - 1) + B^2(2x - 2) \right). \]

The spline curve \( \gamma \) of degree \( m \) is written as a linear combination of shifted B-splines, and we call each coefficient a control point. Then

\[ \gamma(t) = \sum_i p_i B^m(t - i). \]
Notice that each control point $p_i$ only influences $t \in [i, i+m]$. To simplify our notation let $p$ be the column vector of control points and let $B(t)$ be the row vector of the translates of $B^m$. Thus $\gamma$ is written $\gamma(t) = B(t)p$. The refinement equation (1.6) becomes $B(t) = B(2t)S$, where $S$ is a matrix with entries

$$S_{2i+k,k} = \frac{1}{2^{m-1}} \binom{m}{k}.$$ 

This is refinement of the basis elements. Therefore, we can write $\gamma$ as a linear combination of B-splines $B(2t)$ and new control points $Sp$

$$\gamma(t) = B(t)p = B(2t)Sp.$$ 

This shows that basis refinement corresponds to control point refinement. Repeating the refinement gives us

$$\gamma(t) = B(t)p^0$$

$$= B(2t)p^1 = B(2t)S^0p$$

$$\vdots$$

$$= B(2^jt)p^j = B(2^jt)S^j p^0$$

Thus the relationship given between control points at level $j$ of subdivision is $p^{j+1} = Sp^j$. Now consider the control points themselves. The graph of the control points with lines connecting consecutive points is a piecewise linear approximation of the curve $\gamma$. The initial control points $p$ are a coarse approximation, and each multiplication by $S$ gives us a finer approximation. For the purposes of computer visualization, one draws the piecewise linear curve to the desired level of detail, instead of drawing the curve $\gamma$.

### 4-8 Subdivision

Subdivision schemes on surfaces are defined depending on the rule of partitioning and the goal is to ensure higher smoothness of the resulting surface. Usually, this is done
through considering well-known smooth bell-shaped functions \( \varphi \) to be a generating function for the regular schemes, meaning that the limiting surface \( \gamma \) will have the representation

\[
\gamma(x) = \sum_k p_k \varphi(x - k),
\]

where \( p_k \) are the initial values at the control points. Often the generating function of choice is a box spline. For instance, three-directional quartic box splines lead to the Loop subdivision scheme [12], and the tensor product biquadratic and bicubic splines lead to Doo-Sabin [9] and Catmull-Clark [7] subdivision respectively. The box splines are defined similar to the B-splines above. Let \([d_1 \ldots d_m]\) be a set of directions. The box spline \( B^m(x), x \in \mathbb{R}^2 \) can be computed using the following recurrence:

\[
B^j(x) = \int B^{j-1}(x - t d_j) dt,
\]

with \( B^0 \) being the delta function. The directions can be repeated and usually three or four directions are chosen. The directions for four-directional box splines are \((1, 0), (0, 1), (1, 1), \) and \((1, -1)\). The 4-8 subdivision scheme uses bisection refinement as the one in newest vertex bisection defined in Section 1.3 but with the additional requirement: if an internal edge is bisected, then both its adjacent triangles are bisected. For the unit square the usual initial partition consists of four triangles received by adding both diagonal of the square and declaring their intersection point to be the newest vertex for all four triangles (it can be received from the initial partition in Figure 1.1 by subdividing its internal edge). On each step we subdivide all the triangles to receive the next partition. The signature property of this partition is that the valence (number of edges meeting at a vertex) of the internal vertices is either 4 in case of newest vertices, or 8 otherwise. This is the reason for the name “4-8 subdivision scheme”. Full description of the 4-8 subdivision scheme is given in [15]. Here we show only the schemes for the regular internal vertices noting that there are special rules for treating the boundary vertices, as well as the cases of extraordinary
vertices which can be introduced in the initial partition with valences different from 4 or 8. The mask for calculating the value at the new vertex at the middle of the diagonal is given in Figure 1.5. The masks for updating the values at the existing vertices are given in Figure 1.6. Note that in both figures the partitions are shown before the new bisections are made.

The generating function for the 4-8 subdivision scheme on a regular partition is the four-directional box spline for which each of the four directions is taken twice. It is a piecewise polynomial function of degree 6 and has continuous derivatives of order 4. Furthermore, the subdivision surfaces produced by the scheme are $C^4$ continuous almost everywhere, except at extraordinary vertices where they are $C^1$ continuous.
CHAPTER 2

ASSIMILATION AND PROCESSING OF POINT CLOUDS

This chapter builds on methods from Learning Theory to develop techniques and algorithms for handling the specific and difficult case of irregular point clouds. The main idea is to take methods that have have proven to be successful for relatively regular point clouds and extend them to handle irregular point clouds. We preserve the efficacy of these methods by transforming our point cloud to a different topology, one in which the point cloud is semi regular. In particular, the point clouds received from stationary LiDAR are more regular when represented in spherical coordinates. Even in spherical coordinates, there still could be some gaps in the data where the LiDAR did not receive a return. The sensor location is the origin, and the points are given in angle-angle-distance. Indeed, most LiDAR hardware naturally outputs the readings in spherical coordinates, only to later be transformed to Cartesian coordinates to assimilate with existing data sets. Here we only consider point clouds resulting from a LiDAR scans, and do not consider dense and regular (in Cartesian space) point clouds resulting from several scans or perfect knowledge. One area of our investigation is to assess how the Cartesian and spherical coordinate systems interact. Another area is how to diminish the effects of missing data and jumps in the distances from the sensor.

2.1 THEORETICAL FRAMEWORK

The practical algorithms developed in this chapter realize ideas expressed in the work of Binev, Cohen, Dahmen, DeVore, and Temlyakov in *Universal Algorithms*
for Learning Theory. Here we discuss the main results for approximations using piecewise constants [4] and piecewise polynomials [2] over adaptive partitions. Let $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$ be the point cloud. The convergence of the approximation $\tilde{s}$ to the regression function $s_{\rho}$ is usually evaluated in probability or in expectation. The approximation $\tilde{s}$ converges to $s_{\rho}$ in probability if for all $\eta > 0$ the limit of $\text{Prob}\{\|s_{\rho} - \tilde{s}\| \geq \eta\}$ is zero as the sample size approaches infinity. When the limit of $\mathbb{E}(\|s_{\rho} - \tilde{s}\|^2)$ is zero as the sample size approaches infinity, the approximation is said to converge in expectation. Given a partition $P$ of $X$, the piecewise constant approximation $\tilde{s}$ is the solution to the least-squares problem. More specifically, $\tilde{s}$ is the piecewise constant function over $P$ that is the minimizer of

$$\frac{1}{m} \sum_{i=1}^{m} (y^{(i)} - f(x^{(i)}))^2,$$

which implies that, over each cell of $P$, $\tilde{s}$ is simply the empirical average. Their algorithm consists of the following steps:

- Compute the error over each cell.
- Threshold at level $t_m := \kappa \sqrt{\log m / m}$ to obtain the smallest proper subtree of the master tree containing cells with error larger than $t_m$ ($\kappa$ a constant).
- Complete the subtree and obtain the corresponding partition $P$.
- Compute $\tilde{s}$ by empirical risk minimization over $P$ as described above.

Technically, the regression function $s_{\rho}$ needs to be in $A_{\gamma} \cap B^v$. The approximation class $A_{\gamma}$ can be viewed as a smoothness space of order $\gamma > 0$ with smoothness measured with respect to $\rho_X$. Given $v > 0$, $B^v$ is a smoothness space which measures the regularity of a function $f$ by the size of the adaptive partition. Details for the algorithm and the smoothness spaces can be found in [4]. The main result for piecewise constant approximation is the following theorem.
Theorem 2.1. Piecewise Constant Approximation [4] Let \( \beta, \gamma > 0 \) be arbitrary. Then, there exists \( \kappa_0 = \kappa_0(\beta, \gamma, M) \) such that if \( \kappa \geq \kappa_0 \), then whenever \( s_\rho \in A^\gamma \cap B^v \) for some \( v > 0 \), the following concentration estimate holds

\[
\text{Prob} \left\{ \| s_\rho - \tilde{s} \| \geq \tilde{c} \left( \frac{\log m}{m} \right)^{\frac{v}{2v+1}} \right\} \leq C m^{-\beta} \tag{2.1}
\]

as well as the following expectation bound

\[
E \left( \| s_\rho - \tilde{s} \|^2 \right) \leq C \left( \frac{\log m}{m} \right)^{\frac{2v}{2v+1}} \tag{2.2}
\]

where \( \tilde{c} \) and \( C \) are independent of \( m \).

The order of the approximation is optimal save for the logarithmic term. The analog of Theorem 2.1, when replacing piecewise constant with piecewise polynomial, holds with the addition of some regularity conditions on \( \rho_X \). The counterexample, given in Section 3 of [2], shows that the analog is not true without the regularity conditions. The regularity conditions imposed upon the measure are easily satisfied for measures equivalent to Lebesgue measure, but not in the case of gaps in the measure. Details of the conditions can be found in [2]. When \( \rho_X \) is a very irregular measure, the theorem does not imply optimal results for piecewise polynomial approximation. Regardless of the regularity, the algorithm still yields an approximation \( \tilde{s} \).

The empirical risk minimization does not perform well in probability for the counterexample in [2]. Consider \( \rho_X \) to be an irregular measure with high concentration on one side of the subdomain, and low concentration on the other. Given one realization of the point cloud we may only have points on the side with high concentration of measure and no points on the other. However, in another realization the point cloud may have a few points on the side with low concentration. In both instances the algorithm would approximate the surface, but would receive significantly different results. This is the essence of the counterexample showing the instability of the approximation. Our practical algorithm uses two techniques to suppress the instability.
1. Truncated SVD: The truncated singular value decomposition (SVD) is used in the vertex value calculation because it automatically approximates over both uniformly distributed data and data lying on a hyperplane of lower dimension (see, Section 1.3). Essentially, this finds a subspace on which the regularity conditions hold, and ignores the part that might cause problems.

2. Special Subdivision Criterion: We impose a subdivision criterion that splits a triangle if the distance between the center of mass of the subdomain and the center of mass of the points in the subdomain is relatively large (see, Section 2.3). This criterion reduces the number of triangles in the adaptive partition that may have this issue.

Notice that the first technique modifies the vertex approximation, and the second technique modifies the adaptive partition.

Often the surface represented by the LiDAR point cloud is discontinuous. The discontinuities arise whenever one object ends, from the vantage point of the sensor. Examples are the corner of a building or the outline of a tree or telephone pole. Over these discontinuities, large jumps in the distance from the sensor occur over small changes in the angle. If modeled by a continuous surface, then there would be long skinny triangles along the line-of-sight of the sensor. This hinders the quality of our approximation, so we relax the conformity and continuity requirements in two ways. The first is that we remove empty triangles from the partition. Where there are no points, there is no information from which to construct the surface. The second is that we permit a vertex to have more than one value near discontinuities.

2.2 Spherical Learning Setting and Notation

This section sets the stage of both the spherical and Cartesian coordinate systems in which we process point clouds. The LiDAR sensor location is fixed and rotates to scan
over the range $-\pi/2$ to $\pi/2$ latitudinally and $-\pi$ to $\pi$ longitudinally. Note that some devices are not capable of rotating the full range. Let $\mathbb{R}^3 = X \times Y$ refer to the standard Cartesian coordinate system with $x_1, x_2$ and $y$. Define $S = A \times R$ to be the spherical coordinate system with $\theta, \phi$, and $r$ where $A = \{(\theta, \phi) : \theta \in [-\pi, \pi], \phi \in [-\pi/2, \pi/2]\}$ and $R \subset \mathbb{R}$. We have that $-\pi \leq \theta \leq \pi$ is the longitude, $-\pi/2 \leq \phi \leq \pi/2$ is the latitude, and $r > 0$ is the distance from the origin, i.e. sensor location. Recall the Cartesian coordinates for a point $p$ are of the form $(x, y)$ with $x \in \mathbb{R}^2$ and $y \in \mathbb{R}$. Recall from Preliminaries 1.1 $x$ is referred to as the location and $p = (x, y)$ is a point. In spherical coordinates $p = (a, r)$ where $a = (\theta, \phi)$ is the direction from the origin (sensor) and $r = r(a)$ is the distance to the point from the origin in the direction $a$ and is considered to be a function of $\theta$ and $\phi$. The angle $\theta$ is the angle of rotation about the vertical axis ($y$ direction) from the positive $x_1$ axis. The angle $\phi$ is the angle of elevation from the $x_1$ and $x_2$ plane. The standard conversion formulas from $S$ to $\mathbb{R}^3$ are:

\[
\begin{align*}
x_1 &= r \cos(\phi) \cos(\theta) \\
x_2 &= r \cos(\phi) \sin(\theta) \\
y &= r \sin(\phi)
\end{align*}
\]  

(2.3)

and from $\mathbb{R}^3$ to $S$:

\[
\begin{align*}
r &= \sqrt{|x|^2 + y^2} \\
\phi &= \arcsin \left( \frac{y}{r} \right) \\
\theta &= \text{atan2}(x_2, x_1)
\end{align*}
\]  

(2.4)
More specifically

\[
\text{atan2}(x_2, x_1) = \begin{cases} 
\arctan(x_2/x_1) & x_1 > 0 \\
\arctan(x_2/x_1) + \pi & x_1 < 0, x_2 \geq 0 \\
\arctan(x_2/x_1) - \pi & x_1 > 0, x_2 < 0 \\
\pi/2 & x_1 = 0, x_2 > 0 \\
-\pi/2 & x_1 = 0, x_2 < 0 \\
\text{undefined} & x_1 = 0, x_2 = 0
\end{cases}
\]

is a common function used in computational science to find the angle using inverse tangent and taking into account the quadrant the angle lies in.

We assume that \( D \) consists of independent random observations that are identically distributed according to \( \rho \) on \( A \times R \), and the points in \( D \) can be represented in either coordinate system. In this case the marginal probability measure is denoted \( \rho_A \) and the regression function is

\[
s_\rho(a) = \int_R r d\rho(r|a).
\]

Here we are approximating the function \( s_\rho = s_\rho(a) \). For a spherical triangle \( \Delta_S \subset A \) the quantities that drive our approximation are the following:

- **Spherical Moments:**
  \[
  \int_{\Delta_S} 1 d\rho_A, \int_{\Delta_S} \theta d\rho_A, \int_{\Delta_S} \phi d\rho_A, \int_{\Delta_S} r d\rho_A, \text{ and } \int_{\Delta_S} |r|^2 d\rho_A
  \]

- **Cartesian Moments:**
  \[
  \int_{\Delta_S \times R} x_1 d\rho, \int_{\Delta_S \times R} x_2 d\rho, \int_{\Delta_S \times R} x_1 x_2 d\rho, \int_{\Delta_S \times R} x_1^2 d\rho, \int_{\Delta_S \times R} x_2^2 d\rho
  \]
  \[
  \int_{\Delta_S \times R} x_1 y d\rho, \int_{\Delta_S \times R} x_2 y d\rho, \int_{\Delta_S \times R} y d\rho \text{ and } \int_{\Delta_S \times R} y^2 d\rho
  \]

- **Extrema:**
  \[
  \max\{r : (a, r) \in \Delta_S\} \text{ and } \min\{r : (a, r) \in \Delta_S\}
  \]
The spherical and Cartesian moments have the desired property of additivity. The extrema are not additive, yet are easily determined. When merging two triangles, the maximum \( r \)-value of the parent is the maximum of the two children maxima, and the same for the minimum.

2.3 Surface Approximation

Adaptive Partition

The first step in building our surface approximation is to partition the domain. To arrive at our adaptive partition \( P_A \) we follow the procedures outlined in the preliminaries. Since the point cloud is regular in spherical coordinates, we partition the \( S \) domain, \( A \), which is a square encompassing the point cloud, using newest vertex bisection with force-splitting as explained in Section 1.3 to some finest level. The triangles forming a partition in \( A \) correspond to spherical triangles in \( \mathbb{R}^3 \) where the edges are partial great circles.

There is one difference to note between partitioning \( A \), and the Cartesian domain \( X \). In the preliminaries, the domain \( X \) has clearly defined boundaries, and outside of those boundaries there are no connections. However the triangles on the boundaries of \( A \) with \( |\theta| = \pi \) are neighbors and share vertices or edges. This requires the subdivision scheme to link these triangles and include the connections for the purpose of force-splitting. The connections are also needed for any calculation based on the neighborhood around a vertex.

The second step sorts the points of \( D \) into the triangles of the finest level based on their \( \theta \) and \( \phi \) coordinates. Then the aggregate quantities are calculated on the finest level directly from the points. After this, the algorithm never uses the points themselves. Working back up the master tree, the additive quantities of the children are added together to give the corresponding quantities of the parent. For the max-
imum and minimum, the extrema of the parent is the most extreme extrema of the children. Next we give the criteria which adaptively determines a partition $P_A \in \mathcal{P}$.

The main motivation in deciding our criteria is that we want to have a stable and accurate approximation using the easily computed quantities. Our subdivision criteria marks triangles for subdivision if the variance is large or the center of mass of the triangle is far away from the center of mass of the points it contains. Let $\Delta_S = (v_0, v_1, v_2)$ be a spherical triangle whose vertices have $S$ coordinates $(\theta^{v_0}, \phi^{v_1}, r^{v_0})$. Furthermore let $\{p^{(i)}\}_{i=1}^n$ be the points in $\Delta_S$, with $S$ coordinates $(\theta^{(i)}, \phi^{(i)}, r^{(i)})$. The average $r$-values of the points in $\Delta_S$ is denoted by $\bar{r}$. Similarly, $\bar{\theta}$ and $\bar{\phi}$ denote the averages of $\theta$ and $\phi$. Then the variance of the $r$-values of the points in $\Delta_S$ is

$$\text{Var}_r(\Delta_S) = \frac{\int_{\Delta_S} |r|^2 d\rho_A}{\int_{\Delta_S} 1 d\rho_A} - \left( \frac{\int_{\Delta_S} r d\rho_A}{\int_{\Delta_S} 1 d\rho_A} \right)^2 \approx \frac{1}{n} \sum_{i=1}^n |r^{(i)}|^2 - \bar{r}^2.$$

For the purposes of the subdivision criteria, the center of mass of the points in $\Delta_S$ is $(\bar{\theta}, \bar{\phi})$ and the center of mass of $\Delta_S$ is

$$\left( \frac{\theta^{v_0} + \theta^{v_1} + \theta^{v_2}}{3}, \frac{\phi^{v_0} + \phi^{v_1} + \phi^{v_2}}{3} \right).$$

Define $d(\Delta_S)$ as the Euclidean distance in $S$ between

$$(\bar{\theta}, \bar{\phi}) \quad \text{and} \quad \left( \frac{\theta^{v_0} + \theta^{v_1} + \theta^{v_2}}{3}, \frac{\phi^{v_0} + \phi^{v_1} + \phi^{v_2}}{3} \right).$$

Therefore, a triangle $\Delta_S$ is marked for subdivision if either $\text{Var}_r(\Delta_S)$ or $d(\Delta_S)$ is larger than some respective threshold. We list the specific criteria and thresholds used in the numerical results in Section 2.5.

**Vertex Approximation**

In this section we calculate the $r$-values for the vertices in the adaptive partition $P_A$ generated in Section 2.3 using the aggregate quantities of the points in the triangles.

First we calculate all the local $r$-values for a vertex, where each local value uses only the quantities from one triangle containing the vertex. Then we combine the local $r$-values into a single global $r$-value. Under certain circumstances, we may assign two distinct $r$-values to a single vertex.
Local Approximation

The idea of this section is to give an algorithm to approximate the value of a vertex \( v \) over a single subdomain \( \Delta_S \) that contains \( v \). We use the method of least squares to fit a plane over the triangle \( \Delta_S \) based on the aggregate quantities. Higher degree polynomials are not used because their calculation requires denser point clouds than what we receive from real world LiDAR data subjected to a relatively low sampling rate. Therefore we transition from \( S \) to the real space \( \mathbb{R}^3 \). We cannot simply express the vertices in \( \mathbb{R}^3 \), as they are in the form \((\theta, \phi)\) with unknown \( r \). Until the \( r \)-values are found, each vertex \( v \) could be thought of as a ray emanating from the origin and denoted by \( \vec{v} \).

First we find the plane of best fit, \( K \), through the points in \( \Delta_S = (\vec{v}_0, \vec{v}_1, \vec{v}_2) \) using \( \mathbb{R}^3 \) coordinates. Then the intersection of \( K \) and \( \vec{v}_i \) provides \( r_i \) for each \( i = 0, 1, 2 \). In the case that \( K \) and \( \vec{v}_i \) are almost parallel, the \( r \)-values lie outside of the range of the \( r \)-values of the points in \( \Delta_S \). If so, we clamp the \( r \)-values to the maximum or minimum.

Let \( F = \{(x, y) \in D : x \in \Delta_S\} \) and \( n \) denote the cardinality of \( F \). Then we have the local points \( F = \{x^i, y^i\}_{i=1}^n \). Specifically, we let \( E = \{x^i\}_{i=1}^n \) be the location of the points in \( F \). We can treat the data \((y)\) as both a vector of values, \( y = [y^1, ..., y^n] \), and as a real valued function of the discrete domain \( E \), where \( y(x^i) = y^i \) for \( i = 1, ..., n \).

Let \( f \) and \( g \) be two real valued discrete functions defined on \( E \). Then we define the discrete inner product in the usual way

\[
\langle f, g \rangle_F = \frac{1}{|F|} \sum_{x \in E} f(x)g(x) = \frac{1}{n} \sum_{i=1}^n f(x^i)g(x^i). \tag{2.5}
\]

This discrete inner product relates to the following inner product for continuous functions \( f \) and \( g \) over \( \Delta_S \) in \( S \).

\[
\langle f, g \rangle_{\Delta_S} = \frac{1}{\int_{\Delta_S \times \mathbb{R}} 1 d\rho} \int_{\Delta_S \times \mathbb{R}} f(x)g(x) d\rho.
\]

The inner products are calculated from the Cartesian moments for \( \Delta_S \). We define our linear least squares method that approximates \( s_{\rho}(v) \) using inner products.
Here we describe the least squares setup. Let \( K(F, \mathbf{x}) \) be the linear approximation of at \( \mathbf{x} \). Traditionally, \( K \) is of the form

\[
K(F, \mathbf{x}) := K(F, \mathbf{x}; c_0, c_1, \ldots, c_d) = \sum_{j=0}^{d} c_j \phi_j(\mathbf{x})
\]

where the \( c_j \) are \( d + 1 \) adjustable parameters of \( K \), and the \( \phi_j \) are a basis for linear functions over the domain. Start by letting \( \phi_0(\mathbf{x}) = 1 \). Then the remaining linear functions \( \phi_j \) \((j = 1, \ldots, d)\) are chosen to be orthogonal to \( \phi_0 \). So we set

\[
\phi_j(\mathbf{x}_k) = (x^k_j - \bar{x}_j) \quad \text{for} \quad j = 1, \ldots, d \quad \text{and} \quad k = 1, \ldots, n,
\]

where \( x^k_j \) is the \( j \)th component of \( \mathbf{x}^k \) and \( \bar{x}_j \) is the average of the \( j \)th component of \( \mathbf{x}^i \) for \( i = 1, \ldots, n \). Notice

\[
\langle \phi_0, \phi_j \rangle_F = \langle 1, \phi_j \rangle_F = \frac{1}{n} \sum_{i=1}^{n} (x^i_j - \bar{x}_j) = \frac{1}{n} \sum_{i=1}^{n} (x^i_j) - \bar{x}_j = \bar{x}_j - \bar{x}_j = 0.
\]

Let \( \mathbf{c} = [c_1, c_2, \ldots, c_d] \), \( \bar{\mathbf{x}} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n] \), and \( \bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \). Then our approximation more specifically takes the form

\[
K(F, \mathbf{x}) = c_0 + \sum_{j=1}^{d} c_j (x_j - \bar{x}_j) = c_0 + \mathbf{c} \cdot (\mathbf{x} - \bar{\mathbf{x}}).
\]

This form highlights that \( K(F, \mathbf{x}) \) is a constant approximation with adjustments that increase in magnitude as \( \mathbf{x} \) moves away from the center of mass. The coefficients \( \mathbf{c} \) determine the orientation or tilt of the plane \( K \).

The goal of least squares is to minimize \( \langle K(F, \mathbf{x}) - y, K(F, \mathbf{x}) - y \rangle_{\Delta_S} \), which is achieved at the solution to the following system of equations using the Gramian of the \( \phi_j \) for \( j = 0, \ldots, d \).

\[
\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
0 & \langle \phi_1, \phi_1 \rangle_F & \langle \phi_1, \phi_2 \rangle_F & \cdots & \langle \phi_1, \phi_d \rangle_F \\
0 & \langle \phi_2, \phi_1 \rangle_F & \langle \phi_2, \phi_2 \rangle_F & \cdots & \langle \phi_2, \phi_d \rangle_F \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \langle \phi_d, \phi_1 \rangle_F & \langle \phi_d, \phi_2 \rangle_F & \cdots & \langle \phi_d, \phi_d \rangle_F
\end{bmatrix}
\begin{bmatrix}
c_0 \\
c_1 \\
\vdots \\
c_m
\end{bmatrix}
=
\begin{bmatrix}
\bar{y} \\
\langle y, \phi_1 \rangle_F \\
\langle y, \phi_2 \rangle_F \\
\vdots \\
\langle y, \phi_d \rangle_F
\end{bmatrix}
\]
It is clear that $c_0 = \overline{y}$. Let us denote the real symmetric matrix $A$ and vector $b$ as follows

$$A = \begin{bmatrix}
\langle \phi_1, \phi_1 \rangle_F & \cdots & \langle \phi_1, \phi_d \rangle_F \\
\vdots & \ddots & \vdots \\
\langle \phi_d, \phi_1 \rangle_F & \cdots & \langle \phi_d, \phi_d \rangle_F
\end{bmatrix}$$

and

$$b = \begin{bmatrix}
\langle y, \phi_1 \rangle_F \\
\langle y, \phi_2 \rangle_F \\
\vdots \\
\langle y, \phi_d \rangle_F
\end{bmatrix}.$$

Then we solve $Ac = b$ for $c$. As long as $A$ is nonsingular, we can choose from several methods to solve this type of system. Caution must be taken when $|A|$ is near 0, as our approximation blows up. If the measure was Lebesgue, there would be no issues with the matrix $A$. Missing data could cause the points in $\Delta_S$ to be concentrated in a small region away from $v$. Also, the data could lie on a lower dimensional space causing $|A|$ to be close to zero. The truncated SVD method described in Section 1.3 allows us to process the system $\tilde{A}c = b$ in all cases and model the appropriate geometry. The plane of best fit through the points $F$ is

$$K(F, x) = \overline{y} + c \cdot (x - \overline{x}).$$

Consider the case when the points $F$ are sampled from a power line (Figure 2.1). The system $Ac = b$ is singular and the data in $A$ has two singular values near or equal to zero. We can still solve the truncated system $\tilde{A}c = b$ for $c$. The rotation matrices $U$ and $V$ have been truncated to $U_1$ ($3 \times 1$) and $V_1$ ($3 \times 1$), so the rotation of the solution $c$ is limited. Thus, the dimensionality of the points $F$ only affects the tilt of the plane $K$.

In addition to the issue occurring when the determinant of $A$ is close to 0, we also have a problem when $(x - \overline{x})$ is large. The further away the $x$ is from the center
of mass of $F$, the less reliable our approximation becomes. Therefore a subdivision criterion limiting the distance between the center of mass of $F$ and the center of mass of $\Delta_S$ is added when creating the partition.

Now we find the radius value $r_\ell$ at the intersection of $\vec{v}_i$ and $K$ for $i=0,1,2$. Let $\vec{v} = (\theta, \phi)$ be any ray emanating from the origin. Then using the conversion formulas (2.3) and (2.4) we can write

$$
\vec{v} = \begin{bmatrix}
r_\ell \cos(\phi) \cos(\theta) \\
r_\ell \cos(\phi) \sin(\theta) \\
r_\ell \sin(\phi)
\end{bmatrix}
$$

for any $r_\ell > 0$.

Plugging these values in for $K(F, x)$ we have the following equation to solve for $r_\ell$:

$$
r_\ell \sin(\phi) = \bar{y} + c_1(r_\ell \cos(\phi) \cos(\theta) - \bar{x}_1) + c_2(r_\ell \cos(\phi) \sin(\theta) - \bar{x}_2). 
$$

(2.6)

Then the intersection of the plane of best fit $K$ and the $\vec{v}$ is

$$
r_\ell = \frac{\bar{y} - c_1 \bar{x}_1 - c_2 \bar{x}_2}{\sin(\phi) - c_1 \cos(\phi) \cos(\theta) - c_2 \cos(\phi) \sin(\theta)}.
$$

(2.7)

The value $r_\ell$ is the local value of $v$ for a triangle $\Delta_S$. Hence we write $r_\ell(v, \Delta_S)$.

One potential issue is that $K$ and $\vec{v}$ could be close to parallel. Then $r_\ell$ would lie outside the range of the $r$-values in $F$. The algorithm in Learning Theory [2] suggests that we clamp the value $r_\ell$ between $r_M := \max\{r : (\theta, \phi, r) \in F\}$ and
To clarify the meaning of clamp, when \( r_\ell < r_m \) we set \( r_\ell = r_m \). Similarly, if \( r_\ell > r_M \), then \( r_\ell = r_M \).

**Global Approximation and Discontinuities**

In this section we combine all of the local \( r \)-values for a vertex to a single \( r \)-value known as the global \( r \)-value. Each vertex may be contained in several different triangles. Let \( N(v) \) be the set of triangles in \( P_A \) that contain the vertex \( v \). Then we have the local values \( \{ r_\ell(v, \Delta_S) \}_{\Delta_S \in N(v)} \). Note that in our construction of \( P_A \), \( N(v) \) contains at most eight triangles.

For continuous regions of the partition we need to assign a single value to each vertex. This value is known as the global \( r \)-value of the vertex \( v \) and is denoted by \( r(v) \). We calculate \( r(v) \) as a weighted average of the local \( r_\ell(v, \Delta_S) \), where the weights are the sum of barycentric coordinates corresponding to \( v \). Recall that \( \lambda_v(\Delta_S, p) \) is the barycentric coordinate of the point \( p \) in the triangle \( \Delta_S \) corresponding to the vertex \( v \). Define the weight of the triangle \( \Delta_S \) with respect to the vertex \( v \) to be

\[
W(v, \Delta_S) := \sum_{p \in \Delta_S} \lambda_v(\Delta_S, p).
\]

Then the global \( r \)-value is

\[
r(v) = \frac{\sum_{\Delta_S \in N(v)} r_\ell(v, \Delta_S) W(v, \Delta_S)}{\sum_{\Delta_S \in N(v)} W(v, \Delta_S)} .
\]

In our particular application the approximation is not continuous everywhere, so we need to relax the continuity requirement. The global \( r \)-value is a weighted sum and does not approximate jumps well. For LiDAR scans, discontinuities occur when the distance from the scanner changes significantly over a small change in angle. Often this results from two different objects being sensed. Other occasions such as near the horizon are discussed in the remarks found in Section 2.3. If the discontinuity occurred within a triangle, then the variance of the \( r \)-values in the triangle would
be large forcing the triangle to split based on the subdivision criteria. However, it is possible that the discontinuity is near an edge or vertex of the partition. A simplification of this situation would be two triangles $\Delta_1$ and $\Delta_2$ sharing a vertex $v^*$ and the $r$-values of $\Delta_2$ are much larger than the $r$-values in $\Delta_1$ (Figure 2.2). The solution is not just to subdivide further, because the same problem would just occur on a finer scale. Furthermore, both triangles could have small variance with respect to $r$, dense distributions, and have moderate changes in global $r$-values within the respective triangle, causing the triangle not to be marked for subdivision. Therefore our solution adjusts the vertex $r$-value, by allowing the vertex to have two different $r$-values.

We determine the different values of $r$ for a vertex $v$ by clustering the triangles in $N(v)$. For the simplification above, it suffices to separate $N(v^*)$ into $N_1(v^*) = \{\Delta_1\}$ and $N_2(v^*) = \{\Delta_2\}$. The global $r$-value for $v^*$ is a weighted sum of $r_\ell(v^*, \Delta_1)$ and $r_\ell(v^*, \Delta_2)$. Now $v^*$ has one $r$-value $r_\ell(v^*, \Delta_1)$ for $\Delta_1$ and another $r$-value $r_\ell(v^*, \Delta_2)$ for $\Delta_2$. It is possible to separate $N$ into at most eight sets, as eight is the maximum number of triangles possible to contain a single vertex. It is a rare occurrence to need three or more sets to distinguish the objects. Therefore, we choose to separate $N(v)$ into two sets $N_1(v)$ and $N_2(v)$ when $v$ is along a discontinuity. If one wishes to separate into more sets, then the procedure is easily extrapolated. Once we determine the sets, then $r_1(v, N_1)$ and $r_2(v, N_2)$ are calculated by weighted averages just as we did before.
To determine the sets $N_1$ and $N_2$ we march through the triangles in $N$. If $r_\ell(v, \Delta_S)$ is less than the global value $r(v)$, then $\Delta_S \in N_1$. If $r_\ell(v, \Delta_S)$ is greater than $r(v)$, then $\Delta_S \in N_2$. If $r_\ell(v, \Delta_S) = r(v)$, then $\Delta_S$ is in both $N_1$ and $N_2$. We only use additional values for $r$ when it significantly decreases the variance around $v$. More specifically, we define the variance of $N$ as the sum of the variances of the triangles in $N$. We decide to use these distinct $r$-values for $v$ if

$$\text{Var}(N) - \text{Var}(N_1) - \text{Var}(N_2)$$

is greater than some threshold.

**Algorithm Options and Remarks**

In this section we suggest some possible variations to the algorithm and make some remarks. We first discuss modifications to the subdivision criteria that produce better adaptive partitions, at the expense of computational efficiency. Another option discussed here is to vary the choice of the sets used to approximate the value at a vertex. Finally we give some remarks about special circumstances.

The adaptive partition $P_A$ received is dependent upon the point cloud and the subdivision criteria implemented. There is a wide range of potential criteria. Here we outline some different options for the subdivision criteria that will serve the same purposes. In the description of the main algorithm, a triangle $\Delta_S$ is marked for subdivision if either $\text{Var}_r(\Delta_S)$ or $d(\Delta_S)$ is larger than some respective threshold. Recall that $\text{Var}_r(\Delta_S)$ can be expressed by the aggregate quantities of $\Delta_S$, namely

$$\int_{\Delta_S} 1d\rho_A, \int_{\Delta_S} r d\rho_A, \quad \text{and} \quad \int_{\Delta_S} |r|^2 d\rho_A v.$$ 

Moreover $d(\Delta_S)$ is expressed in terms of

$$\int_{\Delta_S} 1d\rho_A, \int_{\Delta_S} \theta d\rho_A, \quad \text{and} \quad \int_{\Delta_S} \phi d\rho_A.$$
and the \((\theta, \phi)\) coordinates of the vertices of the triangle. One desire is for \(P_A\) to have triangles with reasonable distance between the center of mass of the triangle and the center of mass of the points in the triangle. An alternative would be to measure the distance of the centers of mass in \(\mathbb{R}^3\). The issue here is that the \(x_1\), \(x_2\), and \(y\) coordinates for the triangle vertices are not necessarily calculated. This requires that the \(r\)-values of the vertices have been calculated and converted into \(\mathbb{R}^3\) coordinates. Another possible subdivision criteria is to mark triangles with large changes in the \(r\)-values of the vertices. We do not include this in the main algorithm because it also requires the additional calculation of the \(r\)-values of the vertices at each subdivision level. The benefit of this criterion is that it reduces the number of long skinny triangles received along the line-of-sight of the sensor. In these variations the increased quality is counteracted by the decreased efficiency of the calculations on each level of subdivision. The criteria used in the main algorithm is faster, because only the angles are needed and they are known for all the vertices at the time of subdivision. A similar situation arises with the subdivision criteria requiring the variance to be small. The main algorithm uses the variance of the \(r\)-values of the points in the triangle for the criteria, but one could use the Cartesian coordinates. Using just the variance of \(y\) or any single dimension in \(\mathbb{R}^3\) as the criteria is not helpful because within one triangle any dimension could have a large variance. We resolve this by using the smallest eigenvalue of the Cartesian coordinate covariance matrix as a value describing the variance in \(\mathbb{R}^3\). The additional calculation here is in finding the eigenvalue.

When calculating the vertex \(r\)-value of a vertex \(v\), a plane of best fit is found over some region containing \(v\). Moreover, the main algorithm calculates a local \(r_L\)-value over the triangles containing \(v\) and then a global value as a weighted average of the local values. An alternative is to fit the plane to a different neighborhood around \(v\). Let \(U\) denote the neighborhood around \(v\). Letting \(U\) equal \(X\) is too general and we
would lose the local topology around \( v \). One idea: in the regions where the surface is continuous, we let \( U = N(v) \) and calculate the plane of best fit over \( U \). In this case there would be no need for separate local and global calculations of \( r \). For the discontinuous regions the set \( U \) could be clustered into \( U_1 \) and \( U_2 \) as discussed in 2.3. Then fit the plane to each \( U_1 \) and \( U_2 \) and assign \( v \) the two resulting \( r \)-values. The downside to \( U = N(v) \) is that our approximation would be poor when \( v \) is topologically similar to a local maximum or minimum. Other options are possible for the choice of \( U \) and would affect the \( r \)-values and quantities computations accordingly.

Now we discuss some issues that may appear in our resulting surface approximation. The first is referred to as the horizon issue. A horizon is characterized by a flat surface nearly parallel with the sensor that extends far away from the sensor. A significant change in the radius may occur over a small change in the angle, forcing the triangles near a horizon to split because \( \text{Var}_r(\Delta S) \) would be large. There are too few points over the large gap to conclude with any certainty that the surface is continuous and flat. Thus, we choose not to modify this criterion. Another issue is that the approximation of the vertex \( r \)-value is most accurate and stable at the center of mass of the neighborhood \( N(v) \). Thus we have the option to add a post processing step of moving interior vertices to the center of mass of the points contained in \( N(v) \). For clustered vertices, we split \( v \) into two vertices. One copy moves to the center of mass of \( N_1(v) \) and the other vertex copy moves to the center of mass of \( N_2(v) \).

2.4 Progressive Transmission of an Approximation

We shift now to the problem of transmission of data. Not only do we want a quality surface approximation, but we also want an efficient encoding method. The multiresolution structure inherent in the newest vertex bisection method allows for varying levels of detail. The point cloud is considered to be very large and direct transmission of the entire point cloud is not reasonable. An example of the need for compression
is the case of approximating an unknown terrain. A robot or an unmanned vehicle may collect data, then transmit the point cloud information back to base. This transmission may progressively transmit the data. At first the robot may send very few bits, and then sends more as the user requests. Multiresolution method provides the framework for our encoding, and we introduce nonlinearity in the multiresolution process itself. In our representation we give higher priority to certain geometric features (e.g. local extrema) by switching them with less significant elements in the multiresolution tree. We start by explaining our tree structure and then describe the switching process.

**Vertex Tree Structure**

Our encoding will be based on a vertex tree instead of a triangle tree structure. The idea is to build a tree to store information for the vertices. Let $V$ be a tree where each node is a vertex. The initial partition $P_0$ has two triangles which form a *diamond*. Force-splitting the *diamond* results in the addition of a single new vertex $v^*$. Then we let the root of $V$ be $v^*$. Splitting all of the edges opposite $v^*$ results in four new vertices, namely $v_1, v_2, v_3,$ and $v_4$. Define the children of $v^*$ to be $v_1, v_2, v_3,$ and $v_4$. Then we recursively define the children of a vertex $v$ to be the new vertices added as a result of splitting the edges opposite $v$. Triangles of the partition are right-isocoleses, so the smallest angle is $\pi/4$. This implies that no vertex can be contained in more than eight triangles. Therefore every internal node of the tree $V$ has no more than four children.

The tree $V$ is very repetitive. Any vertex internal to the partition will appear twice, because of the force-splitting procedure. Whenever the partition is conformal, we have no need for this repetition. However, in the spherical approximation, some vertices have two distinct values and would make use of the duplication of vertices in $V$. So we define $V'$ to be the tree of vertices with root $v^*$ and when a new vertex
Figure 2.3 The vertex $v^*$ is created when the black diagonal edge $E$ is subdivided. The red edges result from the subdivision of $E$. The vertices $v'_1$ and $v'_2$ lie on $E$, and the vertices $v_1$ and $v_2$ are the vertices opposite $E$.

is created, it is appended to the tree as the child of the vertex that initiated the subdivision. Thus $V'$ is a pruned subtree of $V$ with no duplicate vertices. Whenever the subdivision is uniform, the tree $V'$ will be extremely lopsided with the majority of vertices on the left side. For uniform subdivisions, the tree structures for $V, V'$ and $T$ are fixed base only on the dimensions of the initial square $X$. Therefore, our encoding of the surface will be based on these trees.

**Correction Value**

The location values of the vertices are automatically encoded within the structure of the vertex tree. Now we need to encode the $y$ (or $r$) values of the vertices. Better than simply encoding the whole value, we predict a value of the vertex based on the four vertices of older generations (coarser level) around it. Then we only encode the difference, which we call the correction value. The correction bits are the binary digits of the correction value. We label these older generation vertices $v'_1, v'_2, v_1, \text{and } v_2$ as seen in Figure 2.3.

Let $C(v^*)$ be the correction value of a vertex $v^*$, and $\hat{y}(v^*)$ be the predicted value of $v^*$. Then $C(v^*) = y(v^*) - \hat{y}(v^*)$. The prediction is based on the $y$-values of the vertices $v_1, v_2, v'_1 \text{ and } v'_2$. The edge subdivided in the creation of $v^*$ is the
edge from $v_1'$ to $v_2'$. We let $Max = \max \{(y(v_1), y(v_2), y(v_1'), y(v_2'))\}$ and $Min = \min \{(y(v_1), y(v_2), y(v_1'), y(v_2'))\}$.

We have the following three variants of the prediction formula:

Line Average Prediction

$$\tilde{y}(v^*) := \left\lfloor \frac{1}{2} (y(v_1') + y(v_2')) \right\rfloor ; \quad (2.9)$$

Total Average Prediction

$$\tilde{y}(v^*) := \left\lfloor \frac{1}{4} (y(v_1) + y(v_2) + y(v_1') + y(v_2')) \right\rfloor ; \quad (2.10)$$

Max and Min Average Prediction

$$\tilde{y}(v^*) := \left\lfloor \frac{1}{2} (Max + Min) \right\rfloor . \quad (2.11)$$

Vertices on the boundary will not have the vertex $v_2$ in the calculation of the predicted value. To modify the prediction formula for boundary vertices we remove $v_2$ from all of the formulas and from the calculation of the $Max$ and $Min$. Then in Equation (2.10) we multiply by $1/3$ instead of $1/2$. The formula for calculating the prediction value is not limited to the three we described here. These were chosen as simple predictions that are easily computed. For example, one could choose to use something similar to a plane of best fit through the vertices. Equations (2.9) and (2.10) are motivated by the $L^2$ error, and (2.11) is motivated by the $L^\infty$ error.

**Switches to Preserve Local Extrema**

In the standard multiresolution process, each stage of the progressive transmission would be the information corresponding to some subtree of $V$. Then the vertices or triangles not contained in the subtree would be approximated based on vertices in the subtree. One issue is that an important vertex may be present at a finer level not present in the subtree. To resolve this we introduce nonlinearity to the multiresolution process by switching the $y$-values of finer level important vertices with the $y$-values
of coarser vertices. For our purposes local extrema are considered important, as we aim to preform well in the Hausdorff error. This results in the coarsest level \((P_0)\) containing the global maximum and minimum \(y\)-values. As we refine, the previously switched vertices will be unswitched.

Here we elaborate on the term switch. Vertices consist of a location and a \(y\)-value; however, in \(V\) vertices behave more like placeholders of the correction bits. All of the encoded information for \(v\) is stored in its spot in \(V\). Switching the \(y\)-value of two vertices \(v_a\) and \(v_b\) is to switch the locations in \(V\). So the correction bits for \(v_a\) will be stored in the location of \(v_b\) in \(V\), and vice-versa. Furthermore the switch only occurs between vertices along the same line. One reason we do not switch with the other vertices, is that the generation gap between the vertices on the line is one where the gap is larger for the other vertices. Another reason is that the direction of the switch can be encoded with only one bit when there are only two choices. Let \(v'_{M'}\) be the vertex with the larger \(y\)-value along the same edge as \(v^*\), and \(v'_{m}\) be the smaller of the two.

We assume we already have a uniform partition to the finest level with the \(y\)-values of the vertices either given or approximated. Starting from the finest level we merge all of the triangles level by level until the initial partition is reached. Since our partitions are conformal we merge diamonds and boundary triangles instead of individual triangles. Each merge reduces the vertices in \(V\) by one and the triangles in \(T\) by one or two. The \(y\)-values of vertices are switched only when the vertex to remove is a local extrema. Recall from the prediction formulas the vertices \(v^*, v_1, v_2, v'_1, v'_2\), and the values \(Max\) and \(Min\). Now \(v^*\) is the vertex to be removed instead of added. If \(y(v^*) \in [Min, Max]\), then \(v^*\) is not considered a local extrema. In which case, the vertex values will not be switched, and everything proceeds as normal. However, if \(y(v^*)\) is outside the range of \(Max\) and \(Min\) then \(v^*\) is considered a local extrema and we switch the \(y\)-values. More specifically, when \(y(v^*) > Max\) the values \(y(v^*)\) and
$y(v'_M)$ are switched. If $y(v^*) < Min$ then the values $y(v^*)$ and $y(v'_m)$ are switched.

**Encoding and Decoding**

Progressive transmission of our surface approximation requires the following information to be sent: the triangle tree structure $T$, the vertex tree structure $V$, switches and correction values. Encoding the trees $T$ and $V$ only requires the dimensions of $X$ and the finest resolution. The finest resolution is the length of the smallest edge in the finest partition, which is typically one. Therefore the header includes the dimensions of $X$ and the finest resolution. Other information included in the header is the four $y$-values of the initial partition and the index of the maximum correction bit. Note that the $y$-values of the initial partition contain the global maximum and minimum due to the switches. The correction values are sent bit by bit, so the index of the maximum bit allows us to know the magnitude of the first group of correction bits sent. We estimate the number of bits in the header to be about 84.

The number of bits in the maximum correction value determines how many transmissions will be sent. We refer to a transmission as a wave. The $i$th wave sends the information vertices with the $i$th correction bit nonzero and all of their ancestors. The initial information for a vertex in $V$ is encoded in either two or four bits. Consider the initial information for $v$. The first bit indicates whether or not $v$ was switched, where 0 implies not switched and 1 means switched. If the first bit is one, then the second indicates which vertex $v$ switched with. Recall that $v$ could only be switched along the edge, so 0 and 1 are sufficient to indicate the vertex. On the other hand if the first bit is 0, then the second bit indicates whether or not we sent a correction bit for this vertex. If the first two bits are 00 this means that no information about $v$ nor any vertex below $v$ in $V$ is sent, and is called a zero-tree. Signifying a zero-tree tremendously reduces the number bits sent on each wave, especially with the early waves. If not a zero-tree, then the third and fourth bits are sent. The third bit
indicates the sign of the correction value and the fourth bit is the $i$th correction bit. Note that the initial bits for vertices far down the vertex tree $V$ are often sent in a later wave. For $v$ the waves after the one with the initial bits will only contain the corresponding correction bit and nothing else.

2.5 Numerical Results

Virginia Tech Data

Andrew Kurdila, Department of Mechanical Engineering & Unmanned Systems, Virginia Tech and his team provided the three point clouds analyzed in this chapter. Each point cloud was created from two SICK Commercial LiDAR scanners (www.sick.com) mounted on a vertically rotating motor, which the team registered and combined to form the data sets (Figure 2.4). The SICK scanners are used by ARO Multidisciplinary University Research Initiative (MURI) and USAT. The scanners are considered stationary. The point cloud VT was created from two stationary scans of the courtyard of Hancock Hall, Virginia Tech in connection with MURI Topic # 28, Dynamic Modeling of 3D Urban Terrain, which one can access at the following website: http://imi.cas.sc.edu/MURIwebsite/aro-broad-agency-announcement.

The 81,150 points making up the point cloud VT are provided in spherical coordinates (longitude, latitude, distance). The color scheme for the spherical point cloud and approximations range from green to red as the latitude increases and blue
is added as the distance from the sensor increases. This scheme results in an intuitive visualization of the spherical point cloud. Standard coloring schemes color according to the $r$-value, $z$-value, or the intensity value given in the LiDAR scan. The detailed RGB color is

$\left( \frac{\phi + \pi/2}{\pi}, 1 - \frac{\phi + \pi/2}{\pi}, \frac{r}{\max\{r\}} \right)$

for the coordinate $(\theta, \phi, r)$.

Figures 2.5 and 2.9 are different views of the original VT point cloud consisting of 81,150 points, and are both distortions between Cartesian space and spherical space. The other figures are of the approximation generated by the algorithm in this chapter. Figures 2.10 and 2.11 are of particular interest as they highlight the phenomenon that occurs along the line-of-sight from the sensor. The objects sensed in VT include three trees, a building and a hallway. The algorithm was executed with the following parameters from Section 2.3:

- Splitting Criteria: Triangles with $\text{Var}_r(\Delta_S) > 0.8$ or $d(\Delta_S) > 0.5$ are subdivided.

- A vertex will have two distinct $r$-values when

$$\text{Var}(N) - \text{Var}(N_1) - \text{Var}(N_2) > 3.$$

**Compression Results**

In this section we present the results of our compression on a 2049x2049 STM (standard terrain map) file of the Grand Canyon. The maximum height value is 1709 meters and the minimum is 1336 meters. Hausdorff error is reported in meters. We compare our results in Tables 2.2, 2.3 and 2.4 with that of JPEG 2000 using the Hausdorff error, which is found in Table 2.1. Naively we send our bits directly and do not compress the bitstream itself. The team who developed JPEG 2000 invested time and effort on the compression of their bitstream. Therefore, it is not appropriate to
directly compare our method and JPEG 2000. We provide the following as ballpark measurements of performance. Our results show that for extreme compression and full recovery our encoding outperforms JPEG 2000. For intermediate Hausdorff errors between 3 and 10, JPEG 2000 is slightly better. JPEG 2000 was not particularly geared for Hausdorff error, and we are not surprised when it does not perform well in Hausdorff error for the extremes.

Table 2.1 Wavelet Results (JPEG 2000)

<table>
<thead>
<tr>
<th>Bits</th>
<th>Original Bits</th>
<th>Compression</th>
<th>Hausdorff Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2576</td>
<td>67174416</td>
<td>.00383%</td>
</tr>
<tr>
<td>2</td>
<td>14024</td>
<td>67174416</td>
<td>.02088%</td>
</tr>
<tr>
<td>3</td>
<td>61512</td>
<td>67174416</td>
<td>.09157%</td>
</tr>
<tr>
<td>4</td>
<td>80912</td>
<td>67174416</td>
<td>.12045%</td>
</tr>
</tbody>
</table>
Figure 2.9 Notice the occluded points behind the large tree along the line-of-sight.

Figure 2.10 Long skinny triangles occasionally appear along the line-of-sight.

Figure 2.11 Triangles with radial distance larger than 2 are further subdivided, so there are much fewer skinny triangles along the line-of-sight.

Figure 2.12 Moved the interior vertices to their respective centers of mass to improve the accuracy of the approximation.

Table 2.2 Line Average Prediction

<table>
<thead>
<tr>
<th>Triangulation</th>
<th>Bits</th>
<th>Original Bits</th>
<th>Compression</th>
<th>Hausdorff Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>390</td>
<td>67174416</td>
<td>0.0058%</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>2834</td>
<td>67174416</td>
<td>0.00422%</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>18008</td>
<td>67174416</td>
<td>0.02681%</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>75689</td>
<td>67174416</td>
<td>0.11268%</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>309623</td>
<td>67174416</td>
<td>0.46092%</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1136636</td>
<td>67174416</td>
<td>1.69207%</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>3358479</td>
<td>67174416</td>
<td>4.99964%</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>7648647</td>
<td>67174416</td>
<td>11.38625%</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>14620231</td>
<td>67174416</td>
<td>21.76458%</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>26335126</td>
<td>67174416</td>
<td>39.20410%</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 2.3  Total Average Prediction

<table>
<thead>
<tr>
<th>Triangulation</th>
<th>Bits</th>
<th>Original Bits</th>
<th>Compression</th>
<th>Hausdorff Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>150</td>
<td>67174416</td>
<td>.00022%</td>
<td>27</td>
</tr>
<tr>
<td>2</td>
<td>1700</td>
<td>67174416</td>
<td>.00253%</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>12168</td>
<td>67174416</td>
<td>.01811%</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>53434</td>
<td>67174416</td>
<td>.07955%</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>221580</td>
<td>67174416</td>
<td>.32986%</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>862531</td>
<td>67174416</td>
<td>1.28402%</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2775789</td>
<td>67174416</td>
<td>4.13221%</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>6801281</td>
<td>67174416</td>
<td>10.12481%</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>13858297</td>
<td>67174416</td>
<td>20.63032%</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>25767964</td>
<td>67174416</td>
<td>38.35979%</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2.4  Max and Min Average Prediction

<table>
<thead>
<tr>
<th>Triangulation</th>
<th>Bits</th>
<th>Original Bits</th>
<th>Compression</th>
<th>Hausdorff Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>242</td>
<td>67174416</td>
<td>.00036%</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>1637</td>
<td>67174416</td>
<td>.00244%</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>13473</td>
<td>67174416</td>
<td>.02006%</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>62768</td>
<td>67174416</td>
<td>.09344%</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>260524</td>
<td>67174416</td>
<td>.38783%</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>1013978</td>
<td>67174416</td>
<td>1.50947%</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>3163858</td>
<td>67174416</td>
<td>4.70992%</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>7454856</td>
<td>67174416</td>
<td>11.09776%</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>14528905</td>
<td>67174416</td>
<td>21.62863%</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>26218411</td>
<td>67174416</td>
<td>39.03035%</td>
<td>0</td>
</tr>
</tbody>
</table>
CHAPTER 3

DISTRIBUTION-DEPENDENT SUBDIVISION SCHEMES

For a point cloud $D$, the points are divided into subdomains (cells) based on the partition. To improve the computational cost, we wish to only have the aggregate quantities $Q$ representing the points in each subdomain. There are two types of quantities in $Q$: distribution-dependent $Q_X$ and value-dependent $Q_Y$. The distribution-dependent quantities depend only on the location and distribution of the points in $D$. The value-dependent quantities involve the $y$-value of the points in $D$. When a cell of the partition is subdivided, $Q$ of the children needs to be calculated based solely on the $Q$ of the parent and the neighbors of the parent. The general idea of subdivision schemes is to find the function values of finer vertices using the values of vertices from coarser levels. Unique to our algorithm is that our quantities are both distribution-dependent and value-dependent. The method described in this chapter addresses the issues of irregular sampling and gaps in point clouds. Traditional methods operate under the assumption that a single cell from subdivision has Lebesgue measure, however we wish to learn the measure within each cell as well as the function value. One particular benefit of learning the measure is that cells on the boundary are treated the same as cells with neighbors of zero measure. No special scheme is needed for the boundary cells.

We first approximate the distribution-dependent quantities $Q_X$ then use them to construct a polynomial approximation of the value-dependent quantity $Q_Y$. The foundation for this approach is explained over the intervals ($d = 1$) in Section 3.1. Then in Section 3.3 we discuss the algorithm over triangles ($d = 2$).
Recall that $D$ is a point cloud with distribution $\rho$, and marginal probability measure $\rho_X$. Over an interval $I \in X$, let

$$Q_X(I, \rho_X) = (M_0(I, \rho_X), M_1(I, \rho_X), M_2(I, \rho_X)) = \left( \int_I 1 d\rho_X, \int_I x d\rho_X, \int_I x^2 d\rho_X \right)$$

$$Q_Y(I, \rho_X) = M_y(I, \rho_X) = \int_I y d\rho_X. \quad (3.1)$$

For a triangle $\Delta \in X$, let

$$Q_X(\Delta, \rho_X) = (M_{00}(\Delta, \rho_X), M_{10}(\Delta, \rho_X), M_{01}(\Delta, \rho_X)) = \left( \int_{\Delta} 1 d\rho_X, \int_{\Delta} x_1 d\rho_X, \int_{\Delta} x_2 d\rho_X \right)$$

$$Q_Y(\Delta, \rho_X) = M_y(\Delta, \rho_X) = \int_{\Delta} y d\rho_X. \quad (3.2)$$

Optionally, one could use the Bernstein polynomials instead of the traditional moments and let

$$Q_X(I, \rho_X) = \left( \int_I B_2^0(x) d\rho_X, \int_I B_2^1(x) d\rho_X, \int_I B_2^2(x) d\rho_X \right) \quad (3.5)$$

and

$$Q_X(\Delta, \rho_X) = \left( \int_{\Delta} B_{110}^2(x) d\rho_X, \int_{\Delta} B_{101}^2(x) d\rho_X, \int_{\Delta} B_{011}^2(x) d\rho_X \right). \quad (3.6)$$

Note that the quantities based on the Bernstein polynomials are also additive. The use of the Bernstein polynomials increases the numerical stability. Furthermore, the Bernstein polynomials are simply a change of bases from the standard polynomial bases. So, one may transform the quantities between the two bases as desired. To simplify the explanations we use the moments in the standard bases for the distribution-dependent quantities in the descriptions and calculations of Sections 3.1 - 3.3.
Since $\rho_X$ is unknown, we approximate the moments by summing over the points in each cell. For the points $\{(x_i, y_i)\}_{i=1}^n$ in the interval $I$ we have

$$Q_X(I, \rho_X) = \left( n, \sum_{i=1}^n x_i, \sum_{i=1}^n (x_i)^2 \right)$$  \hspace{1cm} (3.7)

$$Q_Y(I, \rho_X) = \sum_{i=1}^n y_i,$$  \hspace{1cm} (3.8)

and for the points $\{(x_1^i, x_2^i, y^i)\}_{i=1}^m$ in the triangle $\Delta$ we have

$$Q_X(\Delta, \rho_X) = \left( m, \sum_{i=1}^m x_1^{(i)}, \sum_{i=1}^m x_2^{(i)} \right)$$  \hspace{1cm} (3.9)

$$Q_Y(\Delta, \rho_X) = \sum_{i=1}^m y^{(i)}.$$  \hspace{1cm} (3.10)

### 3.1 Approximation of $Q$ over intervals

**Approximation of the Distribution-Dependent Quantities**

Standard subdivision schemes reproduce the polynomials on each subdivision step. As an analogy to polynomial reproduction, at each step we want to reproduce the moments for certain types of measures. Given the moments of an unknown measure $\rho$ over some region $R$, we wish to find a known measure $\mu$ such that $Q_X(R, \rho) = Q_X(R, \mu)$, and we say that $\mu$ represents $\rho$. Then we approximate the distribution-dependent quantities of the children with respect to $\rho$ by the direct calculation of the quantities with respect to the known measure $\mu$. The representative measures we consider are parameter-dependent measures so they can easily be determined.

First consider probability measures $\mu$ of the form $d\mu = \omega(x)dx$ with:

- **Polynomial Weight Measures**:

  $$\omega(x) = \alpha + \beta x + \gamma x^2$$

  The parameters of this type of measure are $(\alpha, \beta, \gamma)$. Since $\mu$ is a probability measure, there is a constraint equation which defines $\alpha$ in terms of $\beta$ and $\gamma$. 

55
• Subdomain Measures:

\[ \omega(x) = \frac{\chi_{[a,b]}}{b-a} \quad \text{or} \quad \omega(x) = \frac{(1 - \chi_{[a,b]})}{2 - b + a}. \]

The parameters of this type of measure are \((a, b)\). These measures are Lebesgue measures of either \([a, b]\) or \([-1, a] \cup [b, 1]\) for \(-1 \leq a \leq b \leq 1\) and single point measures when the Lebesgue measure is zero.

Once we establish the schemes over these types of probability measures, then the results are extended by introducing a constant \(C\) such that \(\mu\) is now of the form \(d\mu = C\omega(x)dx\). The main idea of the method does not change with the addition of the constant, there is just one more parameter to calculate before we know the measure. In both cases above, the additional calculation is simple. Our final scheme blends the two schemes together because each scheme on its own does not make for a good method, but together they perform well. Polynomial weight measures do not have the range to be able to represent all possible measures. On the other hand, we show that measures can always be represented by a subdomain measure, but they introduce a gap for any measure not exactly Lebesgue on the entire interval. This is unlikely to be realistic for measures close to Lebesgue. Thus, we combine the two schemes into one blended scheme.

We describe our schemes over the interval \([-1,1]\) as there is an affine transformation to and from a general interval and the interval \([-1,1]\). Thus, the parent interval is \([-1,1]\) and is divided into its left child \([-1,0]\) and its right child \([0,1]\). Here we describe the details of the transformation. Any element from a general interval \([r,s]\) can be transformed to the standard interval \([-1,1]\) by the affine transformation

\[ T(x) = \frac{2}{s-r} x + \frac{r+s}{r-s}, \]

whose inverse is

\[ T^{-1}(x) = \frac{2}{s-r} \left( x - \frac{r+s}{2} \right). \]
Using the transformation $T$ we also transform $Q_X([r, s])$ to $Q_X([-1, 1])$.

\begin{align*}
M_0([-1, 1]) &= \frac{2}{s - r} M_0([r, s]) \\ M_1([-1, 1]) &= \frac{4}{(s - r)^2} M_1([r, s]) - \frac{2(r + s)}{(s - r)^2} M_0([r, s]) \\ M_2([-1, 1]) &= \frac{8}{(s - r)^3} M_2([r, s]) - \frac{8(r + s)}{(s - r)^3} M_1([r, s]) - \frac{2(r + s)^2}{(r - s)^3} M_0([r, s]) .
\end{align*}

(3.11) (3.12) (3.13)

If Bernstein polynomials are used for the moments instead of $1, x, \text{ and } x^2$, then the quantities would be affine invariant and this transformation would not be needed. We assume during the development of our algorithm that the interval is $[-1, 1]$, then using the transformations above, arrive at the quantities over the general interval.

**Description of the Geometry of the Moments**

Here we establish the setup for our geometric reasonings. Let $\mu$ be a probability measure, then $M_0([-1, 1], \mu) = 1$. Define $u(\mu) := M_1([-1, 1], \mu)$ and $v(\mu) := M_2([-1, 1], \mu)$. Thus the quantities $Q_X([-1, 1], \mu) = (1, u(\mu), v(\mu))$ and each measure $\mu$ corresponds to a vector $q(\mu) := (u(\mu), v(\mu)) \in \mathbb{R}^2$. We refer to this coordinate system as the $uv$-plane. The children’s quantities can also be expressed in the $uv$-plane by restricting $\mu$ to $[-1, 0]$ or $[0, 1]$. Then $Q_X([-1, 0], \mu) = Q_X([-1, 1], \mu|_{[-1,0]})$.

For the purpose of the $uv$-plane we ensure it is a probability measure by dividing $M_0([-1, 0], \mu)$. In fact, for any measure $\mu$ we define

$$u(\mu) := \frac{M_1([-1, 1], \mu)}{M_0([-1, 1], \mu)} \quad \text{and} \quad v(\mu) := \frac{M_2([-1, 1], \mu)}{M_0([-1, 1], \mu)},$$

and again $q(\mu) = ((u(\mu), v(\mu))$. The vector $q$ is not unique to the measure $\mu$. In fact, there are infinitely many probability measures such that the corresponding $q$ is the same vector. From the parent’s vector in the $uv$-plane, our algorithm approximates the children’s $uv$-vector.

Now we explore the geometric properties of the $uv$-plane and the coordinates of $q(\mu)$. The Dirac delta function can be considered a probability measure concentrated
at a single point \( x \) in the interval. We refer to these delta functions as single point measures \( \delta_x \) for some \( x \in [-1, 1] \). Then we have that \( Q_X([-1, 1], \delta_x) = (1, x, x^2) \), so \( u(\delta_x) = x \) and \( v(\delta_x) = x^2 \). Let the curve \( g \) be defined by \( g(u) = u^2 \) for all \( u \in [-1, 1] \) in the \( uv \)-plane. Then \( q(\delta_x) = (u(\delta_x), g(u(\delta_x))) \) for all single-point measures \( \delta_x \). We define \( \Omega \) to be the convex hull of \( g \). Then the vector \( q \) for any probability measure is contained in \( \Omega \). Loosely speaking any probability measure can be considered as a convex combination of the corresponding single-point measures.

The main idea on the interval \([-1, 1]\) is to use the \textit{distribution-dependent} quantities \( Q_X([-1, 1], \rho) \) and the geometric behavior in the \( uv \)-plane to approximate \( Q_X([-1, 0], \rho) \) and \( Q_X([-1, 0], \rho) \). The \textit{distribution-dependent} quantities are very local quantities, so \( Q_X([-1, 1], \rho) \) are the only quantities used to find \( Q_X([-1, 0], \rho) \) and \( Q_X([0, 1], \rho) \). To do this a one-to-one correspondence is established between the measure parameters and the appropriate region in the \( uv \)-plane. However, an explicit formula may not be known, in which case we develop a scheme for finding \((u, v)\) of the children.
Polynomial Weight Measures

In this section we wish to reproduce polynomial weight measures $\eta$ of the form $d\eta = C\omega(x)dx$ where $\omega(x) = \alpha + \beta x + \gamma x^2$. We first develop the method for probability measures, then calculate the constant $C$. Considering $\eta$ a probability measure of the form $d\eta = \omega(x)dx$ produces two properties:

- $\int_{-1}^{1} (\alpha + \beta x + \gamma x^2) \, dx = 1$
- $\omega(x) = \alpha + \beta x + \gamma x^2 \geq 0$ for all $x \in [-1, 1]$.

Thus

$$\alpha = \frac{1}{2} - \frac{1}{3}\gamma. \quad (3.14)$$

Ensuring that the measure is nonnegative produces more restrictions on $\alpha, \beta$, and $\gamma$. Thus $\omega(x) > 0$ for all $x \in [-1, 1]$. More specifically $\omega(0) = \alpha \geq 0$. Then using the equation (3.14) we have $1/2 - 1/3\gamma \geq 0 \Rightarrow \gamma \geq 3/2$. We also have that the endpoints must be nonnegative, so $\alpha + \beta + \gamma \geq 0$ and $\alpha - \beta + \gamma \geq 0$. Adding the two together we have $\alpha + \gamma \geq 0$, which implies that $\gamma \geq -3/4$. We also get that $|\beta| \leq \alpha + \gamma = 1/2 + 2/3\gamma$. To summarize, we have $-3/4 \leq \gamma \leq 3/2$ and $|\beta| \leq 1/2 + 2/3\gamma$.

When $-3/4 \leq \gamma \leq 0$, we have that $\omega$ is either a concave down parabola or a line, so we only need to satisfy the endpoint criteria. Then the current restriction of $\beta$ is sufficient. The restriction on $\beta$ is not sufficient when the $w$ is a concave up parabola with the minimum inside of the interval $[-1, 1]$. Now consider $\gamma$ positive, we have that $\omega$ is a concave up parabola with a minimum at $x = \frac{-\beta}{2\gamma}$. Plugging the constraint on $\alpha$ and the location of the minimum into $\omega$, we have the minimum value $\frac{6\gamma - 4\gamma^2 - 3\beta^2}{12\gamma}$. If $\frac{-\beta}{2\gamma} \in [-1, 1]$, then the minimum value needs to be greater than or equal to 0. So $6\gamma - 4\gamma^2 - 3\beta^2 \geq 0$, which implies $|\beta| \leq \sqrt{2\gamma - \frac{4}{3}\gamma^2}$. In the standard
conic form we have,

$$\frac{4}{3} \beta^2 + \frac{16}{4} \left( \gamma - \frac{3}{4} \right)^2 \leq 1,$$  \hspace{1cm} (3.15)

which is an ellipse centered at $\gamma = \frac{3}{4}$ and $\beta = 0$ with major axis $\frac{3}{4}$ along $\gamma$ and minor axis $\frac{\sqrt{3}}{2}$ along $\beta$.

We now determine which values of $\gamma$ require the constraint (3.15). For $0 < \gamma \leq 3/8$, we have the following string of inequalities

$$\frac{1}{2} + \frac{2}{3} \gamma \geq \sqrt{2\gamma - \frac{4}{3} \gamma^2} \geq 2\gamma.$$  

If $|\beta| > 2\gamma$ we only need the end point criterion $|\beta| \leq \frac{1}{2} + \frac{2}{3} \gamma$ as the minimum is not in $[-1, 1]$. If $|\beta| \leq 2\gamma$, both criteria are already satisfied. So when $-\frac{3}{4} \leq \gamma \leq 3/8$, we have that $|\beta| \leq \frac{1}{2} + \frac{2}{3} \gamma$ is sufficient to ensure $w$ is nonnegative on $[-1, 1]$. The remaining values $3/8 \leq \gamma \leq 3/2$, give us the following string of inequalities

$$2\gamma \geq \frac{1}{2} + \frac{2}{3} \gamma \geq \sqrt{2\gamma - \frac{4}{3} \gamma^2}.$$  

Thus $|\beta|$ is forced to be less than $2\gamma$ by the end point criterion. Then we have that $|\beta| \leq \sqrt{2\gamma - \frac{4}{3} \gamma^2}$ is the appropriate constraint on $\beta$ given $3/8 \leq \gamma \leq 3/2$. Notice that for $\gamma = 3/8$ the value $\sqrt{2\gamma - \frac{4}{3} \gamma^2} = \frac{1}{2} + \frac{2}{3} \gamma$.

To summarize, there are two regions in the parameter space of $\beta$ and $\gamma$. The first region $R_1$ is given by

$$-\frac{3}{4} \leq \gamma \leq \frac{3}{8} \quad \text{and} \quad |\beta| \leq \frac{1}{2} + \frac{2}{3} \gamma.$$  

The second region $R_2$ is given by

$$\frac{3}{8} \leq \gamma \leq \frac{3}{2} \quad \text{and} \quad \frac{4}{3} \beta^2 + \frac{16}{4} \left( \gamma - \frac{3}{4} \right)^2 \leq 1.$$  

In the $uv$-plane we have that

$$u = \int_{-1}^{1} x \omega(x) dx = \frac{2}{3} \beta \quad \text{and} \quad v = \int_{-1}^{1} x^2 \omega(x) dx = \frac{8}{45} \gamma + \frac{1}{3}.$$  \hspace{1cm} (3.16)

60
Figure 3.3 The blue region denotes the region $\Gamma$ in which the vectors corresponding to polynomial weights lie. The vector $(0, 1/3)$ corresponds to Lebesgue measure over the entire interval $[-1,1]$.

So the image of the region $R_1$ in the $uv$-plane is

$$\frac{-1}{5} \leq v \leq \frac{1}{5} \quad \text{and} \quad |u| \leq \frac{5}{2}v - \frac{1}{2}. \quad (3.17)$$

The image of the region $R_2$ in the $uv$-plane is given by

$$\frac{2}{5} \leq v \leq \frac{3}{5} \quad \text{and} \quad 3u^2 + \left(\frac{15}{2}\right)^2 \left(v - \frac{7}{15}\right)^2 \leq 1.$$

We let $\Gamma$ be the region in the $uv$-plane corresponding to the transformations of both $R_1$ and $R_2$. Loosely speaking, $\Gamma$ is the set of $uv$ vectors for polynomial weights. Now consider the shape of the region $\Gamma$. One region is defined by an ellipse centered at $(0, \frac{7}{15})$ with major axis $\frac{\sqrt{3}}{3}$ along $u$ and $\frac{2}{15}$ along $v$. The other region is given by the interior of the absolute value function $v = 2/5|u| + 1/5$. The boundaries of these two regions intersect at $(\pm \frac{1}{2}, \frac{2}{5})$. In fact, the lines $v = 2/5u + 1/5$ and $v = -2/5u + 1/5$ are the tangent lines to the ellipse at the points $(\pm \frac{1}{2}, \frac{2}{5})$. To show this we examine the implicit derivative of the ellipse, which is

$$6u + \frac{15^2}{2} \left(v - \frac{7}{15}\right) \frac{dv}{du} = 0.$$  

This implies that $\frac{dv}{du} = \frac{-4u}{75v - 35}$ and at the points $(\pm \frac{1}{2}, \frac{2}{5})$ we have that $\frac{dv}{du} = \pm \frac{2}{5}$.

The correspondence between $(\beta, \gamma)$ and $(u, v)$ is one-to-one and the inverse relation is

$$\beta = \frac{3}{2}u \quad \text{and} \quad \gamma = \frac{45}{8}v - \frac{15}{8}. \quad (3.18)$$
Now we formulate and prove the result from this section.

**Lemma 3.1.** Let $\rho$ be a measure on $[-1,1]$. If $q(\rho)$ in $\Gamma$, then we construct a polynomial weight measure $\eta$ of the form $d\eta = C\omega(x)dx$ such that $Q_X([-1,1],\rho) = Q_X([-1,1],\eta)$.

**Proof.** The inversion formulas (3.18) give us the explicit values for $\beta$ and $\gamma$. From $\gamma$ we use (3.14) to determine $\alpha$. So far we have

$$u(\rho) = \frac{\int_{-1}^{1} xd\rho}{\int_{-1}^{1} 1d\rho} = \int_{-1}^{1} xw(x)dx$$

and

$$v(\rho) = \frac{\int_{-1}^{1} x^2d\rho}{\int_{-1}^{1} 1d\rho} = \int_{-1}^{1} x^2w(x)dx.$$  \hfill (3.19)

Thus we let $C = \int_{-1}^{1} 1d\rho$. Then $M_0([-1,1],\rho) = C = M_0([-1,1],\eta)$. Furthermore

$$M_1([-1,1],\rho) = u(\rho)M_0([-1,1],\rho) = C\int_{-1}^{1} xw(x)dx = M_1([-1,1],\eta)$$

and finally

$$M_2([-1,1],\rho) = v(\rho)M_0([-1,1],\rho) = C\int_{-1}^{1} x^2w(x)dx = M_2([-1,1],\eta).$$

Therefore, $Q_X([-1,1],\rho) = Q_X([-1,1],\eta)$. \hfill $\square$

Given a measure $\rho$ with $q(\rho)$ in $\Gamma$, we determine the representative measure $\eta$. The representative measure $\eta$ is completely known at this point, and we simply calculate the quantities for the children.

$$Q_X([-1,0],\rho) \approx \left(\int_{-1}^{0} C\omega(x)dx, \int_{-1}^{0} Cx\omega(x)dx, \int_{-1}^{0} Cx^2\omega(x)dx\right)$$  \hfill (3.21)

$$Q_X([0,1],\rho) \approx \left(\int_{0}^{1} C\omega(x)dx, \int_{0}^{1} Cx\omega(x)dx, \int_{0}^{1} Cx^2\omega(x)dx\right)$$  \hfill (3.22)

If the measure $\eta$ is precisely of second degree polynomial form, then our approximation is perfect. Because the acceptable region $\Gamma$ is not equal to $\Omega$, some measures will not have a polynomial weight representation. Thus we need to develop an additional scheme.
Subdomain Measures

In this section we construct a scheme that preserves subdomain measures of the forms $d\mu = C\omega(x)dx$ where

$$\omega(x) = \frac{\chi_{[a,b]}}{b-a} \quad \text{or} \quad \omega(x) = \frac{(1 - \chi_{[a,b]})}{2 - b + a}.$$ 

Measures of the first form are referred to as interval measures and of the second form as complementary measures. As we did for polynomial weight measures, we first develop the method for probability measures, then calculate the constant $C$. The main result from this section is that subdomain measures of these forms create an equivalence class for all measures. Two measures $\rho$ and $\mu$ are in the same equivalence class if $Q_X([-1,1],\rho) = Q_X([-1,1],\mu)$. In other words, given a measure $\rho$, we construct a measure $\mu$ of the form above such that $q(\rho) = q(\mu)$. We then calculate the children’s quantities over $\mu$ as an approximation of the quantities over $\rho$. Once the parameters $C, a$ and $b$ are determined, then we know the measure and thus know the measure of the children. As it turns out, finding $a$ and $b$ is challenging for these measures so we first prove that a one-to-one correspondence exists and then develop a scheme to directly find $(u, v)$ of the children.

First we consider the interval measures $\mu$ of the form $d\mu = \frac{\chi_{[a,b]}}{b-a} dx$ on $[-1,1]$. The possible values for $a$ and $b$ are $-1 \leq a \leq b \leq 1$, which is a triangle in the $ab$-plane with vertices $(-1,1), (-1,-1)$ and $(1,0)$. To clarify, when $a = b$ the measure is concentrated entirely at $a$. In this case, $\mu := \delta_a$. When $a = -1$ and $b = 1$, then $\mu$ is the Lebesgue measure over the entire interval $[-1,1]$. Let $S = \{(a,b) : -1 \leq a < b \leq 1\}$ and let $S^\circ$ denote the interior of $S$. We let $M = M(a,b) = (u(a,b), v(a,b))$ denote the transformation from the coordinates $(a,b) \in S$ to the $uv$-plane by the equations

$$u(a,b) = \int_{-1}^{1} x \, d\mu = \frac{\int_{a}^{b} x \, dx}{b-a} = \frac{a+b}{2} \quad \text{(3.23)}$$
Figure 3.4 On the left we have the set $S$ in the $ab$-plane. On the right is the corresponding region $\Lambda$ in the $uv$-plane. The colors indicate where the region and boundaries are mapped by $M$. The region corresponds to interval measures.

and

$$v(a, b) = \int_{-1}^{1} x^2 \, d\mu = \int_{a}^{b} \frac{x^2 \, dx}{b - a} = \frac{a^2 + ab + b^2}{3}. \quad (3.24)$$

This region in the $uv$-plane is referred to as $\Lambda$, and $M : S \to \Lambda$. Note that $M(a, b)$ corresponds precisely to $q(\mu)$ when $\mu$ is of the form $d\mu = \chi_{[a,b]}(x) \, dx$.

Next we consider the complementary measures $\mu^c$ of the form $d\mu^c = \frac{(1 - \chi_{[a,b]})}{2 - b + a} \, dx$ on $[-1, 1]$. When $a = b$, then $d\mu^c = 1/2dx$. The only limiting case occurs when $(a, b) \in S$ approaches $[-1, 1]$.

**Lemma 3.2.** As $(a, b) \in S$ approaches $[-1, 1]$, $M^c(a, b)$ is on the line $v = 1$ in the $uv$-plane.

**Proof.** Consider the case $a = -1$ and $b$ approaching 1. Then we have that $M^c(-1, b)$ is $u = \frac{1 - b^2}{2(-1 - b + 2)}$, which goes to 1 as $b$ approaches 1, and $v = \frac{1 - b^3}{3(-1 - b + 2)}$, which also goes to 1 as $b$ approaches 1. One the other hand, when $b = 1$ and $a$ approaches $-1$, $M^c(a, 1)$ is $u = \frac{a^2 - 1}{2(a + 1)}$, which goes to $-1$ as $a$ approaches $-1$, and $v = \frac{a^3 + 1}{3(a + 1)}$, which goes to 1 as $a$ approaches $-1$. So the coordinate $u$ depends heavily on the way in which we approach $(-1, 1)$. Now for $\tau, \epsilon > 0$, consider $a = -1 + \epsilon$ and $b = 1 - \tau \epsilon$. As $\epsilon$ approaches 0, then $u$ approaches $\frac{\tau - 1}{\tau + 1}$ and $v$ approaches 1. Notice that $u$ can be any value in $(-1, 1)$ as $\tau$ is an arbitrary nonnegative constant. Therefore, $(a, b) \in S$ approaching $(-1, 1)$ implies that $M^c(a, b)$ is on the line $v = 1$. \qed
Figure 3.5 On the left we have the set $S$ in the $ab$-plane. On the right is the corresponding region $\Upsilon$ in the $uv$-plane. The colors indicate where the region and boundaries are mapped by $M^c$. The region corresponds to complementary measures. The limiting case $(-1,1)$ is mapped to the line $v = 1$ and the hypotenuse $a = b$ is mapped to $(0, 1/3)$.

Recall that $S^o = \{(a, b) : 1 < a < b < 1\}$ is the interior of $S$ in the $ab$-plane. Let $M^c$ be the transformation from the vector $(a, b) \in S^o$ to the $uv$-plane with respect to the complementary measures. Then $M^c(a, b)$ is given by the equations

$$u(a, b) = \int_{-1}^{1} x d\mu^c = \frac{\int_{-1}^{a} x dx + \int_{b}^{1} x dx}{2 - b + a} = \frac{a^2 - b^2}{2(2 - b + a)} \quad (3.25)$$

and

$$v(a, b) = \int_{-1}^{1} x^2 d\mu^c = \frac{\int_{-1}^{a} x^2 dx + \int_{b}^{1} x^2 dx}{2 - b + a} = \frac{2 - b^3 + a^3}{3(2 - b + a)} \quad (3.26)$$

Let $\Upsilon$ be the region given by (3.25) and (3.26) in the $uv$-plane.

To better understand the regions $\Lambda$ and $\Upsilon$ we examine the transformations of the boundaries of $S$. Table 3.1 gives the $M$ and $M^c$ transformations of the boundaries in $S$ to the $uv$-plane. Define the boundary curve $g_2(u) = 1/3(4u^2 - 2|u| + 1)$.

**Lemma 3.3.** $\Lambda$ is bounded by the curves $g(u)$ and $g_2(u)$ in the $uv$-plane. $\Upsilon$ is bounded by $v(u) = 1$ and $g_2(u)$.

**Proof.** By definition of the regions $\Lambda$ and $\Upsilon$, $M(S) = \Lambda$ and $M^c(S^o) = \Upsilon$. Consider the transformations of the boundaries for $S$.

$M(a, 1) = (1/2(a + 1), 1/3(a^2 + a + 1))$, which corresponds to $g_2(u)$ for $u \in [0, 1]$. Then $M(-1, b) = (1/2(b - 1), 1/3(b^2 - b + 1))$, which corresponds to $g_2(u)$.
Table 3.1 Boundary Transformations of $S$ in $ab$-plane to the $uv$-plane

<table>
<thead>
<tr>
<th>$S$ Bounds</th>
<th>Transformation $M$</th>
<th>Image in $uv$-plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b = 1$</td>
<td>$u = \frac{a + 1}{2}$, $v = \frac{a^2 + a + 1}{3}$</td>
<td>$v(u) = \frac{4u^2 - 2u + 1}{3}$, $u \in [0, 1]$</td>
</tr>
<tr>
<td>$a = -1$</td>
<td>$u = \frac{a - 1}{2}$, $v = \frac{a^2 - a + 1}{3}$</td>
<td>$v(u) = \frac{4u^2 + 2u + 1}{3}$, $u \in [-1, 0]$</td>
</tr>
<tr>
<td>$a = b$</td>
<td>$u = 0$, $v = 1/3$</td>
<td>$v(u) = \frac{4u^2 - 2u + 1}{3}$, $(0, 1/3)$</td>
</tr>
</tbody>
</table>

for $u \in [-1, 0]$. Furthermore, $\lim_{b \to a} M(a, b) = (a, a^2)$, which is the curve $g(u)$. The transformation $M$ is continuous; therefore, $M(S) = \Lambda$ is bounded by $g(u)$ and $g_2(u)$ in the $uv$-plane.

Now consider the complement transformation $M^c(a, b) = (u, v)$ where

$$u = \frac{a^2 - b^2}{2(2 - b + a)} \quad \text{and} \quad v = \frac{2 - b^3 + a^3}{3(2 - b + a)}$$

for any $(a, b) \in S^o$. $M^c$ is not defined for any of the boundaries of $S$, so we will examine the limits. By Lemma 3.3 we have that $(a, b) \to [-1, 1]$ gives us $v = 1$ in the $uv$-plane. Then $\lim_{b \to 1} M(a, b) = (1/2(a - 1), 1/3(a^2 - a + 1)$, which corresponds to $g_2(u)$ for $u \in [-1, 0]$. We also have the limit $\lim_{a \to -1} M(a, b) = (1/2(b + 1), 1/3(b^2 + b + 1)$, which corresponds to $g_2(u)$ for $u \in [0, 1]$. And finally $\lim_{b \to a} M(a, b) = (0, 1/3)$. Since $M^c$ is a continuous transformation on $S^o$, we have that the region $M^c(S^o)$ is contained within the region bounded by $g_2$ and $v = 1$. Therefore, $\Upsilon$ is the open region bounded by $g_2$ and $v = 1$. \hfill \Box

**Lemma 3.4.** Let $\rho$ be a probability measure with $q(\rho)$ in the interior of $\Omega$. Then there exists a measure $\mu$ of the form $d\mu = \omega(x)dx$ with

$$\omega(x) = \frac{\chi_{[a,b]}}{b - a} \quad \text{or} \quad \omega(x) = \frac{1 - \chi_{[a,b]}}{2 - b + a}$$
such that \( q(\rho) = q(\mu) \).

**Proof.** From Lemma 3.3 we see that \( \Lambda \cup \Upsilon \) is the interior of \( \Omega \) and that \( \Lambda \cap \Upsilon = \emptyset \). Essentially, we need to show that the transformations \( M \) and \( M^\mathcal{E} \) are invertible because these transformations directly relate the representative measure parameters as a \( uv \)-vector. This further simplifies to showing that the Jacobian of the transformations are nonzero on the domain. The Jacobian of the transformation \( M \) is \( 6(b-a) \), and only equals zero at \( a = b \). Since the Jacobian is nonzero for \( (a,b) \in S \), we have that \( M \) is invertible.

Now consider the transformation \( M^\mathcal{E} \), whose Jacobian is

\[
J = \frac{(b-a)((a-b)^3 - 12ab - 4)}{6(2-b+a)^3}.
\]

We want to show that \( J \) is not zero for \( (a,b) \in S^\mathcal{E} \), so we need only consider the function \( f(a,b) = (a-b)^3 - 12ab - 4 \). Note that \( f(-1,1) \) is zero, but \( (-1,1) \) is not in the set \( S^\mathcal{E} \). We have that the gradient of \( f \) is given by

\[
\nabla(f) = (3(a-b)^2 - 12b) \mathbf{i} + (-3(a-b)^2 - 12a) \mathbf{j},
\]

and is zero at \((-1,1)\). Thus \( f \) has a critical point at \((-1,1)\) and \( f(-1,1) = 0 \). The second derivatives are

\[
f_{aa} = f_{bb} = 6(a-b) \quad \text{and} \quad f_{ab} = 6(b-a - 2),
\]

(3.27)
which are all defined and less than zero for \((a,b) \in S^o\). By the second derivative test, \(f(a,b)\) has a local maximum at \((-1,1)\), and the second derivatives continue to be negative for all \((a,b) \in S^o\). Therefore, \(J\) is nonzero on \(S^o\), implying the transformation \(M^L\) is invertible. Therefore, the transformations over \(\Lambda\) and \(\Upsilon\) are invertible. \(\square\)

In our setup we are given a vector \((u,v)\) and we want to find the corresponding \((a,b)\). So we now know that such relation exists for \((u,v) \in \Omega\). However, there is no explicit formula for the inverse of \(M\) and \(M^L\). Thus, we develop a scheme for finding the \(uv\)-coordinates of the children. First we develop the scheme for vectors \(uv\) on the boundary, then for \(uv\) on the interior of \(\Omega\).

Our main concern is not about the measures corresponding to the boundary, but we include the scheme here for completeness of our algorithm. Denote the \(uv\)-vectors of the children by \(q_L\) and \(q_R\). The first boundary case is when \((u_1,u_1^2)\) is the \(uv\)-vector corresponding to the measure. Then the representative measure is \(\delta_{u_1}\). If \(u_1 \in [-1,0]\), then \(q_L = (u_1,u_1^2)\) and \(q_R = (0,0)\). Similarly, if \(u_1 \in [0,1]\), then \(q_L = (0,0)\) and \(q_R = (u_1,u_1^2)\). The second boundary case occurs when the coordinates are \((u_2,1)\) so \(v = 1\), which corresponds to the limiting case. Then if \(u_2 \in [-1,0]\), we have that \(q_L = (u_2,1)\) and \(q_R = (0,0)\). Similarly, if \(u_2 \in [0,1]\), then \(q_L = (0,0)\) and \(q_R = (u_2,1)\).

The scheme for the interior of \(\Omega\) requires that we further partition the sets \(\Lambda\) and \(\Upsilon\). The additional boundary curves

\[
g_3(u) = \frac{-4|u|^3 + 4u^2\sqrt{u^2 + 4|u|} - 12u^2 + 4|u|\sqrt{u^2 + 4|u|} - 2}{-3|u| + 3\sqrt{u^2 + 4|u|} - 6}
\]

and \(g_1(u) = 4/3u^2\) are needed. As it turns out, these curves distinguish the case \(ab > 0\) from the case \(ab < 0\). The curve \(g_1\) divides \(\Lambda\) and the curve \(g_3\) divides \(\Upsilon\). Essentially, we have one scheme for \(a\) and \(b\) in a single child and another scheme for \(a\) and \(b\) in different children.

We define \(\Lambda_1\) as the region bounded by \(g\), \(g_1\), and \(g_2\) including \(g_1\) from \(-1/2\) to \(1/2\) and \(g_2\) for \(u \in [-1,-1/2] \cup [1/2,1]\). Let \(\Lambda_2\) be the region bounded by \(g_1\) and
Table 3.2  Additional Transformations in the $ab$-plane to the $uv$-plane

<table>
<thead>
<tr>
<th>$ab$-plane</th>
<th>Transformation $M$</th>
<th>Image in $uv$-plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b = 0$</td>
<td>$u = a/2$</td>
<td>$v = a^2/3$, $v = 4/3a^2$, $u \in [-1,0]$</td>
</tr>
<tr>
<td>$a = 0$</td>
<td>$u = b/2$</td>
<td>$v = b^2/3$, $v = 4/3a^2$, $u \in [0,1]$</td>
</tr>
</tbody>
</table>

$g_2$ including $g_2$ from $-1/2$ to $1/2$. Thus, $\Lambda = \Lambda_1 \cup \Lambda_2$. Similarly, define $\Upsilon_1$ to be the open region bounded by $g_2$, and $g_3$. Lastly, define $\Upsilon_2$ as the region bounded by $v(u) = 1, g_2$ and $g_3$ including the curve $g_3$ from $-1/2$ to $1/2$. So $\Upsilon = \Upsilon_1 \cup \Upsilon_2$. For $\Lambda$ in the following lemmas, we divide $S$ into the regions $R_1 = \{ (a,b) \in S : ab \geq 0 \}$ and $R_2 = \{ (a,b) \in S : ab < 0 \}$. For $\Upsilon$, we have $T_1 = \{ (a,b) \in S^o : ab > 0 \}$ and $T_2 = \{ (a,b) \in S^o : ab \leq 0 \}$.

**Lemma 3.5.** $M(R_1) = \Lambda_1$, $M(R_2) = \Lambda_2$, $M^c(T_1) = \Upsilon_1$ and $M^c(T_2) = \Upsilon_2$.

**Proof.** By the definition of $\Lambda$ and $\Upsilon$, we already have that $M(S) = \Lambda$ and $M^c(S^o) = \Upsilon$. Consider the boundary $a = 0$ for all the regions. Then $M(0,b) = (b/2, b^2/3)$ which is $g_1(u)$ for $u \in [0,1]$, and

$$M^c(0,b) = \left( \frac{b^2}{2b-4}, \frac{b^3-2}{3b-6} \right)$$

which is $g_3(u)$ for $u \in [-1,0]$. When $b = 0$ we have that $M(a,0) = (a/2, a^2/3)$ which is $g_1(u)$ for $u \in [-1,0]$, and

$$M^c(a,0) = \left( \frac{a^2}{2a+4}, \frac{a^3+2}{3a+6} \right)$$

which is $g_3(u)$ for $u \in [0,1]$. Recall the boundary transformations in the previous lemma.

Both $M$ and $M^c$ are continuous transformations on their respective domains, so we have that the image of the subregions of the domains is contained in the map of
Figure 3.7 The top left is the set $S$ in the $ab$-plane colored according to the transformation $M$ and the top right is colored according to the transformation $M^c$. The bottom plot is the image in the $uv$-plane of the four regions $\Lambda_1$ (blue), $\Lambda_2$ (pink), $\Upsilon_1$ (green), and $\Upsilon_2$ (yellow).

the boundaries. All that remains is to test one $(a, b)$ transformation for each region. Then $M(1/4,1/2) = (3/8,7/16)$ and $7/16 < g_1(1/4)$ which implies that $M(R_1) = \Lambda_1$. $M(-1/2,1/2) = (0,1/4)$ and $1/4 > g_1(1/2)$ which implies that $M(R_2) = \Lambda_2$.

Next $M^c(1/4,1/2) = (-3/56,121/336)$ and $121/336 < g_3(1/4)$ which implies that $M^c(T_1) = \Upsilon_1$. $M^c(-1/2,1/2) = (0,7/12)$ and $7/12 > g_3(0)$ which implies that $M^c(T_2) = \Upsilon_2$.

Algorithm

Here we describe the nonlinear algorithm we developed to find the $uv$-vectors of the children, given the $uv$-vector of the parent $[-1,1]$. The outline for the algorithm is as follows. Let $\rho$ be any probability measure. We first determine in which region $q(\rho)$
lies. If \( q(\rho) \) is on the boundary of \( \Omega \), then we follow the scheme discussed above. Thus \( q(\rho) \) is in one of the regions \( \Lambda_1, \Lambda_2, \Upsilon_1, \) or \( \Upsilon_2 \). The region indicates whether \( \mu \) is an interval measure or complementary measure and whether or not \( ab \geq 0 \) or \( ab \leq 0 \). Then we wish to find the coordinates \((u, v)\) of the children using the geometry in the \( uv\)-plane. By Lemma 3.4 we know that such a \( \mu \) exists and is unique.

Here are some thoughts before we dive into the technicalities of the algorithm. Let \( q_p = (u_p, v_p) \) be the \( uv\)-coordinates of the parent interval \([-1, 1]\), and let \( q_1 = (u_1, v_1) \) and \( q_2 = (u_2, v_2) \) be the vectors of the children. The parent measure is always a convex combination of the children measures. Hence, \( q_p \) will always lie on the line segment connecting \( q_1 \) and \( q_2 \). The development of the algorithm often uses the connections between probability measure \( \mu \) of the form \( d\mu = \frac{1}{s-r} dx \int_{r}^{s} dx \) and the interval \([r, s]\) itself. So at times we relax our vocabulary in order to get a clear picture of the behavior.

We will exploit the symmetry of the curves and relationships with respect to the \( v\)-axis. This is to be expected because of the symmetry in the transformations. For a probability measure \( \mu \), consider \( q'(\mu) \) where \( q'(\mu) \) is the reflection of \( q(\mu) \) across the line \( u = 0 \). Let \( q_1 \) and \( q_2 \) be the children vectors found from the vector \( q(\mu) \) and let \( q'_1 \) and \( q'_2 \) be their reflections. Then the children vectors for \( q' \) are \( q'_1 \) and \( q'_2 \). Note that \( q'_1 \) now corresponds to the right child and \( q'_2 \) corresponds to the left child. Thus for any measure \( \mu \), we find the children vectors from either \( q(\mu) \) or \( q'(\mu) \). Furthermore, the symmetry of all of the boundary curves and regions implies that \( q(\mu) \) and \( q'(\mu) \) lie in the same region. Therefore, the following algorithms safely assume \( u_p \geq 0 \).

We first consider the simple case of \( u_p = 0 \). Then \( q_p \) lies in either \( \Lambda_2 \), \( \Upsilon_2 \), or \((0, 1/3)\). If \( q_p = (0, 1/3) \), then the corresponding measure \( \mu \) is of the form \( d\mu = \frac{1}{2} dx \). Thus \( q_1 = (-1/2, 1/3) \) and \( q_2 = (1/2, 1/3) \) after readjusting to probability measures. If \( q_p \in \Lambda_2 \), then \( \mu \) is the probability measure corresponding to \([-b, b]\). The left child corresponds to \([-b, 0]\) and the right child corresponds to \([0, b]\). These measures correspond to boundary case \( a = 0 \) or \( b = 0 \), which for \( \Lambda \) is the curve \( g_1 \).
Combining the fact that $u_1 = -u_2$, $q_1$ and $q_2$ lie on $g_1$, and that $q_p$ is on the line segment connecting $q_1$ and $q_2$, we have that $q_1$ and $q_2$ are the intersections of the constant line $v = v_p$ and the boundary curve $g_1$. We have a similar argument for $q_p \in \Upsilon$. The representative measure corresponds to $[-1, -b] \cup [b, 1]$, so the left child corresponds to $[-1, -b]$ and the right child corresponds to $[b, 1]$. Measures of these forms correspond to the boundary curve $g_2$. Thus $q_1$ and $q_2$ will lie on the curve $g_2$. Therefore, $q_1$ and $q_2$ are the intersections of the constant line $v = v_p$ and the boundary curve $g_2$.

Consider the case $q_p \in \Lambda_1$. Then the representative measure corresponds to $[a, b]$ with $ab \leq 0$. This implies that the entire representative measure is contained within only one of the children. Since $u_p > 0$ we have that $q_1 = (0, 0)$ and $q_2 = q_p$.

When $q_p \in \Lambda_2$, the representative measure corresponds to $[a, b]$ with $ab < 0$. Then the left child measure corresponds to $[a, 0]$ and the right corresponds to $[0, b]$. Just as we had before, this implies that $q_1$ and $q_2$ lie on the boundary curve $g_1$. Consider $q_1 = (u_1, v_1)$ and $q_2 = (u_2, v_2)$ in terms of $a$ and $b$. Equations (3.23) and (3.24) give us the following equations:

\[
\begin{align*}
  u_1 &= \frac{a + 0}{2} = \frac{a}{2} \\
  u_2 &= \frac{0 + b}{2} = \frac{b}{2} \\
  v_1 &= \frac{a^2 + 0 \cdot a + 0}{3} = \frac{a^2}{3} \\
  v_2 &= \frac{0 + 0 \cdot b + b^2}{3} = \frac{b^2}{3}.
\end{align*}
\]  

(3.28)  

(3.29)

The line connecting $q_1$ and $q_2$ has slope

\[
\frac{v_2 - v_1}{u_2 - u_1} = \frac{2(b^2 - a^2)}{3(b - a)} = \frac{2(a + b)}{3} = \frac{4}{3} \frac{a + b}{2} = \frac{4}{3} u_p.
\]  

(3.30)

Note that in this case, the slope only depends on the $u$-value of the parent interval. Therefore, $q_1$ and $q_2$ are the intersections of the line through $q_p$ with slope $\frac{4}{3} u_p$ and the curve $g_1$.

Next we consider $q_p \in \Upsilon$. Then we have that the representative measure $\mu$ corresponds to $[-1, a] \cup [b, 1]$ with a gap of length $\ell_p = (b - a)$. Replacing $b$ with $a + \ell_p$ gives
us that $\mu$ corresponds to $[-1, a] \cup [a + \ell_p, 1]$. First we find the gap length by considering the curve $\omega$ given by the $uv$-vector of the measure corresponding to $[-1, t] \cup [t + \ell, 1]$ as $t$ changes. Using equations (3.25) and (3.26) we have that the parametric equations for $\omega$ are given by

$$u(t, \ell) = \frac{t^2 - (t + \ell)^2}{2(2 - (t + \ell) + t)} \quad \text{and} \quad v(t, \ell) = \frac{2 - (t + \ell)^3 + t^3}{3(2 - (t + \ell) + t)}.$$  \hspace{1cm} (3.31)

By solving equation (3.31) for $t$ in terms of $u$ and then substituting into the expression for $v$, we have

$$\omega(u, \ell, [-1, 1]) = \frac{\ell^2}{12} + \frac{\ell}{6} + u^2 + \frac{1}{3} - \frac{2u^2}{\ell}.$$ \hspace{1cm} (3.32)

Generalizing this for the measure corresponding to $[r, t] \cup [t + \ell, s]$, we have that

$$\omega(u, \ell, [r, s]) = \frac{1}{12\ell} (3r^3 + r^2 \ell - 12r^2u + 3r^2s + 12u^2r - 3rs^2 - 2rs\ell - r\ell^2 + 12u^2\ell + 12s^2u - 12u^2s - 3s^3 + s^2\ell + s\ell^2 + \ell^3).$$

Specifically, $q_p$ must lie on the curve $v = \omega(u, \ell, [-1, 1])$ for some $\ell = \ell_p$. Then we find $\ell_p$ by solving $v_p = \omega(u_p, \ell, [-1, 1])$ for $\ell$, which is equivalent to finding the appropriate zero of

$$f(\ell) = \frac{\ell^3}{12} + \frac{\ell^2}{6} + \left(u_p^2 - v_p + \frac{1}{3}\right) \ell - 2u_p^2.$$ \hspace{1cm} (3.33)

using Newton’s method. Note that $\ell_p \in [0, 2]$, $f(0) = -2u^2_p$, and $f(2) = 2 - 2v_p$. Furthermore, $u \in [-1, 1]$ implies that $f(0) \leq 0$ and $v \leq 1$ implies that $f(2) \geq 0$. By the Intermediate Value Theorem, there exists an $\ell_p \in [0, 2]$ such that $f(\ell_p) = 0$.

When $q_p \in \Upsilon_1$ the gap is entirely contained in one child. Since $u_p > 0$, the right child corresponds to the whole interval $[0, 1]$ and the left child corresponds to $[-1, a] \cup [b, 0]$ with the same gap length as the representative measure. Thus $q_2 = (1/2, 1/3)$. For the left child the gap length is known so $q_1$ is the intersection of the curve $\omega(u, \ell_p, [-1, 0])$ and the line through the points $q_p$ and $q_2$.

The final case to consider is $q_p \in \Upsilon_2$. The left child corresponds to $[-1, a]$ and the right child corresponds to $[b, 1]$. So we have that $q_1$ and $q_2$ both lie on the boundary
curve $g_2$ and we also have that they lie on the line passing through $q_p$. In a previous case, we showed there is a very nice relationship between the slope of the line and $u_p$, but we do not have any such relationship here. Therefore, an iterative method is implemented to find the children vectors. The idea is to start with a value for $u_2$ and then determine the corresponding gap length. Once the gap length of the iterative method matches the actual gap we have found the actual value of $u_2$. From $u_2$, then $v_2 = g_2(u_2)$ and $q_1$ is found to be the intersection of the line through $q_2$ and $q_p$ and the curve $q_2$.

The intersections of the curve $g_2$ and the line through the vector $q_p$ and the vector $(0, \omega(0, \ell, [-1, 1]))$ serve as good initial guesses for $u_1$ and $u_2$, which we denote $\tilde{u}_1^{(0)}$ and $\tilde{u}_2^{(0)}$. The $k^{th}$ step of the iteration is denoted $\tilde{u}_1^{(k)}$ and $\tilde{u}_2^{(k)}$. We desire to have the sum of the gap in $[-1, 0]$ and in $[0, 1]$ equal to $\ell_p$. Thus the quality of the approximation at step $k$ is determined by the discrepancy between the gap length of the approximation and $\ell_p$. Let $\tilde{\ell}_1^{(k)}$ be the gap length on $[-1, 0]$ and $\tilde{\ell}_2^{(k)}$ the gap length on $[0, 1]$ based upon the approximation $\tilde{u}^{(k)}$. Fortunately, we do not need to solve $\omega$ for the gap lengths as there is a simpler relationship. Instead we write the gap lengths as a function of $\tilde{u}_1^{(k)}$ and $\tilde{u}_2^{(k)}$. We have that $\ell_p = b - a$, $\tilde{\ell}_1^{(k)} = -\tilde{a}$, and $\tilde{\ell}_2^{(k)} = \tilde{b}$. Furthermore,

$$\tilde{u}_1^{(k)} = 1/2(\tilde{a} - 1), \quad \text{and} \quad \tilde{u}_2^{(k)} = 1/2(\tilde{b} + 1).$$

Thus we write $\tilde{\ell}_1^{(k)} = -2\tilde{u}_1^{(k)} - 1$ and $\tilde{\ell}_2^{(k)} = 2\tilde{u}_2^{(k)} - 1$. The discrepancy at step $k$ is

$$d^{(k)} = \ell_p - (-2\tilde{u}_1^{(k)} - 1 + 2\tilde{u}_2^{(k)} - 1) = \ell_p + 2\tilde{u}_1^{(k)} - 2\tilde{u}_2^{(k)} + 2.$$

Then we adjust by $\tilde{u}_1^{k+1} = \tilde{u}_1^{k} + d^{(k)}/2$. The iteration terminates when $d^{(k)}$ is sufficiently small.

The algorithm so far has only addressed the vectors in $uv$. Let $Q_X([-1, 1], \rho)$ be the quantities supplied for some measure $\rho$. At this point, we need a way of approximating $Q_X([-1, 0], \rho)$ and $Q_X([0, 1], \rho)$ with measures of the form $d\mu = C\omega(x)dx$. 

74
Notice the addition of the constant $C$. The vector corresponding to $\rho$ is
\[
q(\rho) = (u_p, v_p) = \left( \frac{M_1([-1, 1], \rho)}{M_0([-1, 1], \rho)}, \frac{M_2([-1, 1], \rho)}{M_0([-1, 1], \rho)} \right).
\]
From the algorithm above we find approximations for the $uv$-vectors of the children corresponding to an unknown representative measure $\mu$ of the form $d\mu = \omega(x)dx$ with
\[
\omega(x) = \frac{\chi_{[a,b]}}{b-a} \quad \text{or} \quad \omega(x) = \frac{1 - \chi_{[a,b]}}{2 - b + a},
\]
such that $q(\rho) = q(\mu)$. Thus
\[
u_p = \int_{-1}^{1} x\omega(x)dx \quad \text{and} \quad v_p = \int_{-1}^{1} x^2\omega(x)dx
\]
Let $C = M_0([-1, 1], \rho)$. Then $M_0([-1, 1], \rho) = C = M_0([-1, 1], \mu)$. Furthermore,
\[
M_1([-1, 1], \rho) = u_p M_0([-1, 1], \rho) = C \int_{-1}^{1} x\omega(x)dx = M_1([-1, 1], \mu)
\]
and finally
\[
M_2([-1, 1], \rho) = v_p M_0([-1, 1], \rho) = C \int_{-1}^{1} x^2\omega(x)dx = M_2([-1, 1], \mu).
\]
Thus $Q_X([-1, 1], \rho) = Q_X([-1, 1], \mu)$. We know $C$ but we do not explicitly know the parameters $a$ and $b$ for the measure $\mu$. It remains to determine $M_0([-1, 0], \mu)$ and $M_0([0, 1], \mu)$. By definition $M_0([-1, 1], \mu) = M_0([-1, 0], \mu) + M_0([0, 1], \mu)$. Then we write $M_0([-1, 0], \mu) = tC$ and $M_0([0, 1], \mu) = (1 - t)C$ for some parameter $t \in [0, 1]$. Think back to the $uv$-plane with $q(\rho) = q_p$ and children $q_1$ and $q_2$. The closer the vector $q_i$ is to $q_p$, the larger the weight placed on the corresponding child. Let $d$ be the Euclidean distance in the $uv$-plane. Then we define the parameter $t$ as follows
\[
t = \frac{d(q_2, q_p)}{d(q_1, q_2)}.
\]
Therefore, the children distribution-dependent quantities are
\[
M_0([-1, 0], \mu) = \frac{d(q_2, q_p)}{d(q_1, q_2)} C \quad M_0([0, 1], \mu) = \left(1 - \frac{d(q_2, q_p)}{d(q_1, q_2)}\right) C
\]
\[
M_1([-1, 0], \mu) = u_1 M_0([-1, 0], \mu) \quad M_1([0, 1], \mu) = u_2 M_1([0, 1], \mu)
\]
\[
M_2([-1, 0], \mu) = v_1 M_0([-1, 0], \mu) \quad M_2([0, 1], \mu) = v_2 M_1([0, 1], \mu).
\]
Figure 3.8 The overlay of $\Gamma$ from the polynomial weighted scheme and the regions from the subdomain scheme. The blending scheme combines the results from the two schemes into one scheme.

**Blending**

For a measure $\rho$ with quantities $Q_x([-1,1],\rho)$ we use the methods above to find an approximation for the children using either the polynomial weight or subdomain measures. The downside to the polynomial weight scheme is that there are measures that give quantities not possible for the polynomial scheme. That is to say the region $\Gamma$ in $uv$ corresponding to polynomial weight measures does not cover all of $\Omega$.

The polynomial weight scheme can be extended to the regions outside of $\Gamma$, but large errors will occur. To illustrate the errors that arise outside of $\Gamma$, consider the point cloud $D$ that contains 713 points evenly distributed along the interval $[-12,12]$ with a gap from $-7$ to $-2$, and the $y$-values $y = 2x + 1$. Figure 3.9 clearly displays the error that occurs around a gap when the polynomial weight scheme is used for any measure.

On the other hand, all measures can be represented by a subdomain measure. When a measure is close to but not exactly Lebesgue, the representative measure found using the subdomain scheme assumes there is a gap. This is not good, because our point cloud applications will not have perfectly Lebesgue measure even when the
Figure 3.9  The first image depicts the average $y$-values of the 6 initial intervals from which the actual quantities are calculated directly from the point cloud $D$. The other images are levels 1, 2 and 3 of subdivision where the quantities are calculated according to the polynomial scheme described in Section 3.1. These plots illustrate the issue that occurs when strictly using the polynomial scheme. Notice the large errors that occur around the gap.

Lebesgue measure is the most appropriate. Therefore, we choose to blend the two schemes. Let $Q_X([-1, 0], \eta)$ and $Q_X([0, 1], \eta)$ be the quantities calculated from the polynomial scheme and $Q_X([-1, 0], \mu)$ and $Q_X([0, 1], \mu)$ be the quantities from the subdivision scheme.

We approximate the real $Q_X$ with $\tilde{Q}_X$ for some parameter $t$

$$\tilde{Q}_X([-1, 0], \rho) = (1-t)Q_X([-1, 0], \eta) + tQ_X([-1, 0], \mu)$$  (3.34)

$$\tilde{Q}_X([0, 1], \rho) = (1-t)Q_X([0, 1], \eta) + tQ_X([0, 1], \mu).$$  (3.35)

The polynomial approximation is not considered whenever $q(\rho)$ lies outside of $\Gamma$. 

77
Thus, we let $t = 1$ for $q(\rho)$ outside of $\Gamma$. Now assume that $q(\rho) \in \Gamma$. The closer the measure is to Lebesgue, the larger the polynomial approximation weight. In the $uv$-plane, a measure being close to Lebesgue means that the vector $q(\rho)$ is close to $(0, 1/3)$. Consider the line $L(u)$ passing through $q(\rho)$ and $(0, 1/3)$,

$$L(u) = \frac{v(\eta) - 1/3}{u(\eta)} u + \frac{1}{3}. $$

Then $L(u)$ intersects the boundary of $\Gamma$ in two locations $r$ and $s$. Let $d$ be the standard Euclidean distance function. Without loss of generality assume that $d(q(\rho), r) \leq d(q(\rho), s)$. Then we define $t$ to be the ratio of the distances

$$t = \frac{d(q(\rho), (0, 1/3))}{d(r, (0, 1/3))}. $$

For $t \geq 1$, $q(\rho)$ is outside of the $\Gamma$ and we set $t = 1$. Notice that for $q(\rho)$ on the boundary of $\Gamma$, we have that the weight of the polynomial approximation is zero. When $q(\rho) = (0, 1/3)$ the weight for the polynomial function is one.

### 3.2 Approximation of the Value-Dependent Quantities

Now that we have the approximations for distribution-dependent quantities $Q_X([-1, 0])$ and $Q_X([0, 1])$, it remains to find $Q_Y([-1, 0])$ and $Q_Y([0, 1])$. The value-dependent quantities are not as local as the distribution-dependent quantities. Therefore we include the quantities of the neighbor intervals $[-3, -1]$ and $[1, 3]$ in the calculations.

Then we write the children quantities as a linear combination of the parent and neighbor quantities.

$$Q_Y([-1, 0]) = c_0 Q_Y([-3, -1]) + c_1 Q_Y([-1, 1]) + c_2 Q_Y([1, 3]) \quad (3.36) $$

$$Q_Y([0, 1]) = d_0 Q_Y([-3, -1]) + d_1 Q_Y([-1, 1]) + d_2 Q_Y([1, 3]) \quad (3.37) $$

for some coefficients $c_0, c_1, c_2, d_0, d_1,$ and $d_2$. The standard subdivision schemes reproduce polynomials, so we do the same here for the $y$-values. Hence, we use the
test functions $1, x$ and $x^2$ to calculate $Q_Y([-1, 0])$ and $Q_Y([0, 1])$. Equations (3.36), (3.37), and the test functions give us the following systems of equations:

$$
\begin{align*}
\begin{bmatrix}
\int_{-1}^{0} 1d\rho_X \\
\int_{-1}^{0} xd\rho_X \\
\int_{-1}^{0} x^2d\rho_X 
\end{bmatrix}
&= 
\begin{bmatrix}
\int_{-3}^{-1} 1d\rho_X \\
\int_{-3}^{-1} xd\rho_X \\
\int_{-3}^{-1} x^2d\rho_X 
\end{bmatrix}
\begin{bmatrix}
1 \\
1 \\
1 
\end{bmatrix}
\begin{bmatrix}
\int_{1}^{3} 1d\rho_X \\
\int_{1}^{3} xd\rho_X \\
\int_{1}^{3} x^2d\rho_X 
\end{bmatrix}
\begin{bmatrix}
c_0 \\
c_1 \\
c_2 
\end{bmatrix}
\end{align*}
$$

and

$$
\begin{align*}
\begin{bmatrix}
\int_{0}^{1} 1d\rho_X \\
\int_{0}^{1} xd\rho_X \\
\int_{0}^{1} x^2d\rho_X 
\end{bmatrix}
&= 
\begin{bmatrix}
\int_{-3}^{-1} 1d\rho_X \\
\int_{-3}^{-1} xd\rho_X \\
\int_{-3}^{-1} x^2d\rho_X 
\end{bmatrix}
\begin{bmatrix}
1 \\
1 \\
1 
\end{bmatrix}
\begin{bmatrix}
\int_{1}^{3} 1d\rho_X \\
\int_{1}^{3} xd\rho_X \\
\int_{1}^{3} x^2d\rho_X 
\end{bmatrix}
\begin{bmatrix}
d_0 \\
d_1 \\
d_2 
\end{bmatrix}
\end{align*}
$$

We already have an approximation for all of the integrals in these systems. So the systems simplify down to

$$
Q_X([-1, 0]) = 
\begin{bmatrix}
Q_X([-3, -1]) \\
Q_X([-1, 1]) \\
Q_X([1, 3]) 
\end{bmatrix}
\begin{bmatrix}
c_0 \\
c_1 \\
c_2 
\end{bmatrix}
$$

and

$$
Q_X([0, 1]) = 
\begin{bmatrix}
Q_X([-3, -1]) \\
Q_X([-1, 1]) \\
Q_X([1, 3]) 
\end{bmatrix}
\begin{bmatrix}
d_0 \\
d_1 \\
d_2 
\end{bmatrix}
$$

We solve for the coefficients $c_0, c_1, c_2, d_0, d_1, d_2$. At this point we run into the same issue as Section 2.1. The systems could be near singular, causing difficulty directly solving. Therefore, we solve the systems just as before, using the truncated SVD for some threshold. Once the coefficients are found, we plug the values into the equations (3.36) and (3.37), giving us $Q_Y([-1, 0])$ and $Q_Y([0, 1])$. From $Q_Y$ and $Q_X$ an approximation of the surface can be constructed. Our focus is on the calculation and preservation of the quantities, not how the quantities are used in the approximation.
Numerical Results for Distribution-Dependent Subdivision over Intervals

In this section we present the results of the blending algorithm over the intervals. The theory suggests that the algorithm should perform extremely well for quadratic functions and regular distributions with gaps. The domain $X$ is the interval $[-12,12]$ and the initial partition consists of evenly spaced intervals. The intervals are bisected at each level of subdivision. We insert the points from the point cloud into the intervals and directly calculate the distribution-dependent quantities $M_0, M_1, M_2$, and the value-dependent quantity $M_y$. No other part of the algorithm directly accesses the point cloud. Let $I^*$ denote the partition where the quantities are calculated directly from the point cloud. The subsequent levels of subdivision calculate the quantities of the children based on the parent’s and parent’s neighbors’ quantities according to the blending algorithm in Section 3.1. The figures in this section colored magenta are the actual values (not averages). The green intervals are the actual average quantities calculated over the initial partition. Then the blue intervals correspond to the first level of subdivision, red is the second level, and black is the third level of subdivision. At each level we plot $(M_y/M_0), (M_1/M_0), (M_2/M_0)$ values, which correspond to the average $y, x,$ and $x^2$.

For the first example, $I^*$ is six evenly spaced intervals partitioning $X$, and we consider the point cloud $D_1$ which consists of 626 points uniformly distributed from -12 to 12 with gaps $[-9.213, -4.0234]$ and $[3.1, 5.23]$, with $y$-values satisfying

$$y(x) = -3x^2 + 2x + 1.$$ 

Indeed, the results for this situation are great (Figures 3.10 - 3.14). Subdividing to level 10 gives a good idea of the limit function of the subdivision scheme (Figure 3.14).

Expanding the type of functions, our second test function has that $I^*$ is 12 evenly
Figure 3.10  Graphs of the initial averages for $y$, $x$, and $x^2$ calculated from the actual point cloud $D_1$.

Figure 3.11  Level 1 of subdivision: Plots of the averages for $y$, $x$, and $x^2$ calculated only from the quantities of the parent and parent’s neighbors.

Figure 3.12  Level 2 of subdivision: Plots of the averages for $y$, $x$, and $x^2$ calculated only from the quantities of the parent and parent’s neighbors.

Figure 3.13  Level 3 of subdivision: Plots of the averages for $y$, $x$, and $x^2$ calculated only from the quantities of the parent and parent’s neighbors.
Figure 3.14 Comparison of the average $y$-value for level 10 of subdivision (blue) to the actual point cloud $D_1$ (magenta).

spaced intervals over $X$, and the point cloud $D_2$ again consists of 626 points uniformly distributed from -12 to 12 with gaps $[−9.213, −4.0234]$ and $[3.1, 5.23]$. Now we let the $y$-values satisfy

$$y_2 = \sin \left( \frac{x^2}{18} \right).$$

The algorithm performs well for point cloud $D_2$, although not as well as in $D_1$ (Figures 3.15 - 3.19). One interval of interest is $[8, 10] \in I^*$. Notice how the initial interval $[8, 10]$ better approximates the local minimum as we refine strictly from the parent and neighbor values.

3.3 Approximation of $Q$ over Triangles

Now we extend our ideas to $d = 2$ with triangles instead of intervals. We develop our algorithm for the simple case of

$$Q_X(R, \mu) = (M_{00}(R), M_{10}(R), M_{01}(R)) \quad \text{and} \quad Q_Y(R, \mu) = M_Y(R)$$

with

$$M_{00}(R) = \int_R 1 \, d\mu \quad M_{10}(R) = \int_R x_1 \, d\mu$$

$$M_{01}(R) = \int_R x_2 \, d\mu \quad M_Y(R) = \int_R y \, d\mu.$$
Figure 3.15  Graphs of the initial averages for $y$, $x$, and $x^2$ of $D_2$.

Figure 3.16  Graphs of the averages for $y$, $x$, and $x^2$ of $D_2$ at level 1 of subdivision.

Figure 3.17  Graphs of the averages for $y$, $x$, and $x^2$ of $D_2$ at level 2 of subdivision.

Figure 3.18  Graphs of the averages for $y$, $x$, and $x^2$ of $D_2$ at level 3 of subdivision.
Following the same ideas as the $d = 1$ situation, we make the assumption that the parent triangle is $\Delta^* = ((0,1), (-1,0), (1,0))$ and is divided into its left child $\Delta^*_1 = ((0,0), (0,1), (-1,0))$ and its right child $\Delta^*_2 = ((0,0), (1,0), (0,1))$. This assumption is legitimate because there are affine transformations to and from a general right-isosceles triangle to the reference (standard) triangle $\Delta^*$. The transformations are defined here. Let $(b, l, r)$ be a general right-isosceles triangle in $X$. Define $T$ as

$$T = \begin{bmatrix}
1 & 0 & 0 \\
b_1 & l_1 - b_1 & r_1 - b_1 \\
b_2 & l_2 - b_2 & r_2 - b_2
\end{bmatrix} \begin{bmatrix}
1 & 0 & 0 \\
0 & -1 & 1 \\
1 & -1 & -1
\end{bmatrix}^{-1}$$

An element $x^* = (x^*_1, x^*_2)$ of the reference triangle $\Delta^*$ is transformed to $x = (x_1, x_2)$ in the general triangle $(b, l, r)$ by the affine transformation

$$T \begin{bmatrix}
x^*_1 \\
x^*_2
\end{bmatrix} = \begin{bmatrix}
1 \\
x_1 \\
x_2
\end{bmatrix}.$$ 

Thus, elements from the general triangle is transformed to the reference triangle by
the inverse,

\[
T^{-1} \begin{bmatrix}
1 \\
x_1 \\
x_2
\end{bmatrix} = \begin{bmatrix}
1 \\
x_1^* \\
x_2^*
\end{bmatrix}.
\]

Define for a measure \( \mu \) over \( \Delta^* \)

\[
u(\mu) := \frac{M_{10}(\Delta^*, \mu)}{M_{00}(\Delta^*, \mu)} \quad \text{and} \quad v(\mu) := \frac{M_{01}(\Delta^*, \mu)}{M_{00}(\Delta^*, \mu)},
\]

and \( q(\mu) = ((u(\mu), v(\mu)) \). We consider \( u \) and \( v \) as the average values of \( x_1 \) and \( x_2 \) respectively. Thus the vector \( q(\mu) \) will lie in the convex hull of \( \Delta^* \) for measures \( \mu \).

We develop a few methods based on the types of measures we wish to reproduce. The measures we consider are of the form \( d\mu = \omega(x)dx \) with:

- **Weight Measures:**
  \[
  \omega(x) = \alpha \omega_a(x) + \beta \omega_b(x) + \gamma \omega_c(x)
  \]

- **Subdomain Measures:**
  \[
  \omega(x) = \chi_t \quad \text{or} \quad \omega(x) = (1 - \chi_t).
  \]

Once the relationships in the \( uv \)-plane are established, then we introduce a constant \( C \) into the form for the measure. The weight measures have parameters \((\alpha, \beta, \gamma)\) and the subdomain measures have parameters based on the triangle \( t \). The three parent quantities in \( Q_X(\Delta^*) \) are used to determine the parameters. Once the parameters are found, then the children quantities are easily calculated. The last method on the triangle is to combine the two schemes into one blended scheme. The main idea on the triangle \( \Delta^* \) with \( Q_X(\Delta^*) \) is to use the geometric behavior in the \( uv \)-plane to approximate \( Q_X(\Delta_1^*) \) and \( Q_X(\Delta_2^*) \). To do this a one-to-one correspondence is established between the measure parameters and the appropriate region in the \( uv \)-plane. In this case it is possible to algebraically find the inverse.
Weight Measures

Consider the probability measures $\mu$ over the reference triangle $\Delta^*$ such that $d\mu = \omega(x)dx$ where $\omega(x) = \alpha \omega_a(x) + \beta \omega_b(x) + \gamma \omega_c(x)$. We define the weight functions in such a way that

$$\int_{\Delta^*} w_a(x)dx_1dx_2 = \int_{\Delta^*} w_b(x)dx_1dx_2 = \int_{\Delta^*} w_c(x)dx_1dx_2 = 1 .$$

We also impose the following conditions on the weight functions

$$\omega_a((0,1)) = 1 \quad \text{and} \quad \omega_a((-1,0)) = \omega_a((1,0)) = 0 \quad (3.41)$$

$$\omega_b((-1,0)) = 1 \quad \text{and} \quad \omega_b((0,1)) = \omega_b((1,0)) = 0 \quad (3.42)$$

$$\omega_c((1,0)) = 1 \quad \text{and} \quad \omega_c((-1,0)) = \omega_c((0,1)) = 0. \quad (3.43)$$

Therefore the weight functions are defined to be

$$\omega_a(x) = x_2 \quad (3.44)$$

$$\omega_b(x) = \frac{1}{2}(-x_1 - x_2 + 1) \quad (3.45)$$

$$\omega_c(x) = \frac{1}{2}(x_1 - x_2 + 1). \quad (3.46)$$

Since $\mu$ is a probability measure, we have two conditions. The first condition is that $\omega(x)$ is nonnegative for all $x \in \Delta^*$. This implies more specifically that $\omega$ is nonnegative at the vertices of $\Delta^*$. Then we have that

$$\omega((0,1)) = \alpha \geq 0 \quad (3.47)$$

$$\omega((-1,0)) = \beta \geq 0 \quad (3.48)$$

$$\omega((1,0)) = \gamma \geq 0 \quad (3.49)$$

The weight functions $\omega_a, \omega_b,$ and $\omega_c$ are all nonnegative so the inequalities above are sufficient to force $\omega(x) \geq 0$ for all $x \in \Delta^*$. The second condition given by the fact that $\mu$ is a probability measure is that

$$\int_{\Delta^*} \alpha \omega_a(x) + \beta \omega_b(x) + \gamma \omega_c(x)dx = 1 \quad \Rightarrow \quad \frac{1}{3}(\alpha + \beta + \gamma) = 1.$$
Thus \( \alpha = 3 - \beta - \gamma \) making \( \gamma \leq 3 - \beta \). Consider the transformation from the parameters of the weight measures \((\alpha, \beta, \gamma)\) to the moments \((M_{00}, M_{10}, M_{01})\). We simplify this transformation to \((\beta, \gamma)\) to the moments \((M_{10}, M_{01})\) as \(M_{00} = 1\) and \(\alpha\) is a function of \(\beta\) and \(\gamma\).

\[
\begin{align*}
  u &= M_{10} = \int_{\Delta^*} x_1 ((3 - \beta - \gamma) \omega_a(x) + \beta \omega_b(x) + \gamma \omega_c(x)) \, dx \\
  &= \int_{\Delta^*} (3 - \beta - \gamma) x_1 x_2 + \frac{\beta x_1}{2}(-x_1 - x_2 + 1) + \frac{\gamma x_1}{2}(x_1 - x_2 + 1) \, dx \\
  &= \frac{1}{2} \int_{\Delta^*} x_1^2(\gamma - \beta) + x_1 x_2(6 - 3\beta - 3\gamma) + x_1(\beta + \gamma) \, dx \\
  &= \frac{-\beta}{12} + \frac{\gamma}{12} .
\end{align*}
\]

(3.50)

\[
\begin{align*}
  v &= M_{01} = \int_{\Delta^*} x_2 ((3 - \beta - \gamma) \omega_a(x) + \beta \omega_b(x) + \gamma \omega_c(x)) \, dx \\
  &= \int_{\Delta^*} (3 - \beta - \gamma) x_2^2 + \frac{\beta x_2}{2}(-x_1 - x_2 + 1) + \frac{\gamma x_2}{2}(x_1 - x_2 + 1) \, dx \\
  &= \frac{1}{2} \int_{\Delta^*} x_2^2(6 - 3\gamma - 3\beta) + x_1 x_2(\gamma - \beta) + x_2(\beta + \gamma) \, dx \\
  &= \frac{1}{2} - \frac{\beta}{12} - \frac{\gamma}{12} .
\end{align*}
\]

(3.51)

Therefore,

\[(\beta, \gamma) \rightarrow \left(\frac{1}{12}(\gamma - \beta), \frac{1}{12}(6 - \beta - \gamma)\right) .\]

This is a linear relation so the inverse is simply \(\beta = 3 - 6u - 6v\) and \(\gamma = 3 + 6u - 6v\). So we have

\[(u, v) \rightarrow (3 - 6u - 6v, 3 + 6u - 6v) .\]

(3.52)

The conditions on \(\beta\) and \(\gamma\) are that \(\gamma, \beta \geq 0\) and \(\gamma \leq 3 - \beta\). Then \(\beta \geq 0\) implies that \(3 - 6u - 6v \geq 0\), so \(v \leq 1/2 - u\). The inequality \(\gamma \geq 0\) implies that \(3 + 6u - 6v \geq 0\), thus \(v \leq 1/2 + u\). Finally, the inequality \(\gamma \leq 3 - \beta\) gives us that \(v \geq -1/4\). Therefore, the image in the \(uv\)-plane is the triangle \(\Delta_u\) with vertices \((0, 1/2), (-1/4, 1/4)\), and \((1/4, 1/4)\). As to be expected, the weight measures are not enough to cover the range of all possible probability measures.
Here we outline the algorithm and introduce the constant \( C \) into the representative measure \( \eta \). Assume that \( \rho \) is a measure over \( \Delta^* \) such that \( q(\rho) \in \Delta_w \). Then

\[
q(\rho) = \begin{pmatrix}
M_{10}(\Delta^*, \rho) \\
M_{00}(\Delta^*, \rho)
\end{pmatrix}
\begin{pmatrix}
M_{01}(\Delta^*, \rho) \\
M_{00}(\Delta^*, \rho)
\end{pmatrix}^{-1}
\]

From \( q(\rho) \) we have \( \beta = 3 - 6u - 6v, \gamma = 3 + 6u - 6v, \) and \( \alpha = 12v - 3 \) from the inverse equations (3.52), such that

\[
\begin{align*}
u(\rho) &= M_{10}(\Delta^*, \rho) = \int_{\Delta^*} x_1 \omega(x) dx \\
v(\rho) &= M_{01}(\Delta^*, \rho) = \int_{\Delta^*} x_2 \omega(x) dx.
\end{align*}
\]

Let \( C = M_{00}(\Delta^*, \rho) \). Then for the measure \( \eta \) with \( d\eta = C \omega(x) dx \), we have

\[
\begin{align*}
M_{00}(\Delta^*, \rho) &= C = M_{00}(\Delta^*, \eta) \quad (3.53) \\
M_{10}(\Delta^*, \rho) &= u(\rho) M_{00}(\Delta^*, \rho) = C \int_{\Delta^*} x_1 \omega(x) dx = M_{10}(\Delta^*, \eta) \quad (3.54) \\
M_{01}(\Delta^*, \rho) &= v(\rho) M_{00}(\Delta^*, \rho) = C \int_{\Delta^*} x_2 \omega(x) dx = M_{01}(\Delta^*, \eta). \quad (3.55)
\end{align*}
\]

Therefore, from the unknown measure \( \rho \) quantities we construct the known representative measure \( \eta \) such that \( Q_X(\Delta^*, \rho) = Q_X(\Delta^*, \eta) \). Next we directly calculate the quantities of the children over \( \eta \) to approximate the children quantities of \( \rho \). So

\[
Q_X(\Delta^*_1, \rho) \approx Q_X(\Delta^*_1, \eta) \quad \text{and} \quad Q_X(\Delta^*_2, \rho) \approx Q_X(\Delta^*_2, \eta).
\]

88
Subdomain Measures

In this section we develop an algorithm for representing an unknown measure $\rho$ with a known measure $\mu$ of the form $d\mu = C\chi_t dx$ or $d\mu = C(1 - \chi_t)dx$ for some constant $C$ and some triangle $t$. The first form is referred to as triangle measures and the other are complementary measures. Now we define the set $D$ which specifies the types of triangles $t$ we consider. Define three vectors that lie on the boundary of $\Delta^*$ in the $x$-plane as follows:

- $b = (b, 1 - b)$, where $0 \leq b \leq 1$
- $c = (c, 1 + c)$, where $-1 \leq c \leq 0$
- $d = (d, 0)$, where $-1 \leq d \leq 1$

Let $D$ denote the set of triangles of the form $(b, d, (1, 0))$, $(c, d, (-1, 0))$, or $(b, c, (0, 1))$ for any $b$, $c$, and $d$ in the appropriate ranges. Note the symmetry of the triangles in $D$ with respect to the line $x_1 = 0$. The triangle $(b, d, (1, 0))$ with $d > 0$ is the same triangle as $(c, d, (-1, 0))$ with $d < 0$ reflected across the line $x_1 = 0$. In the same way $(b, d, (1, 0))$ with $d < 0$ and $(c, d, (-1, 0))$ with $d > 0$ are symmetric. Therefore, we have $D$ separated into three types of triangles, disregarding the symmetric triangles

- Type 1 (Fig 3.21): $(b, d, (1, 0))$ with $d > 0$

Triangles of Type 1 are defined by the parameters $b, d$ and $C$, and are characterized as triangles contained completely in one child.
• Type 2 (Fig 3.22): \((c, d, (-1, 0))\) with \(d > 0\)

Triangles of Type 2 are defined by the parameters \(c, d\) and \(C\).

• Type 3 (Fig 3.23): \((b, c, (0, 1))\)

Triangles of Type 3 are defined by the parameters \(b, c\) and \(C\).

Type 2 and 3 triangles intersect both children.

**Triangle Measures**

We begin by examining the transformations to the \(uv\)-plane for triangle measures \(\mu\) of the form \(d\mu = C\chi_t dx\) over the different types of triangles \(t\). This section determines the equations for the invertible relationship between the \(uv\)-plane and the parameters \(b, c\) and \(d\), and defines the region of the \(uv\)-plane for which this relationship holds.

We begin by noting that the \(u\) coordinate corresponds to the average \(x_1\) value and the \(v\) coordinate corresponds to the average \(x_2\) value. So the image in the \(uv\)-plane will be contained within the convex hull of \((-1, 0), (1, 0)\) and \((0,1)\).

**Type 1 Triangle Measures** For Type 1 triangle measures, the triangle \(t\) is bounded by the lines

\[
x_1 = 1 - x_2
\]

\[
x_1 = L_{bd}(x_2) = \frac{b - d}{1 - b} x_2 + d
\]

(3.56)

\[
x_2 = 0.
\]
Thus the moments are

\[ M_{00}(\Delta^*, \mu) = \int_{\Delta^*} 1d\mu = \int_t Cdx = \int_0^{b-1} \int_{b\Delta(x_2)}^{1-x_2} Cdx_1dx_2 \]

\[ = \frac{C}{2}(d-1)(b-1) \]

\[ M_{10}(\Delta^*, \mu) = \int_{\Delta^*} x_1d\mu = \int_t Cx_1dx = \int_0^{b-1} \int_{b\Delta(x_2)}^{1-x_2} Cx_1dx_1dx_2 \]

\[ = \frac{C}{6}(d-1)(b-1)(1+b+d) \]

\[ M_{01}(\Delta^*, \mu) = \int_{\Delta^*} x_2d\mu = \int_t Cx_2dx = \int_0^{b-1} \int_{b\Delta(x_2)}^{1-x_2} Cx_2dx_1dx_2 \]

\[ = -\frac{C}{6}(d-1)(b-1)^2. \]

Therefore the transformation \((b, d) \rightarrow (u, v)\) is defined as follows

\[ u(b, d) = \frac{M_{10}(\Delta^*, \mu)}{M_{00}(\Delta^*, \mu)} = \frac{1+b+d}{3} \]

and

\[ v(b, d) = \frac{M_{01}(\Delta^*, \mu)}{M_{00}(\Delta^*, \mu)} = \frac{1-b}{3} \]

The transformation is linear, so the inverse \((u, v) \rightarrow (b, d)\) is easily determined to be

\[ b(u, v) = 1 - 3v \quad \text{and} \quad d(u, v) = -2 + 3v + 3u. \]

Let \(R_1\) denote the image of the square \(0 \leq b \leq 1\) and \(0 \leq d \leq 1\) in the \(uv\)-plane. The

<table>
<thead>
<tr>
<th>Boundary in (bd)-plane</th>
<th>Transformation</th>
<th>Image in (uv)-plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b = 0)</td>
<td>(u = 1/3(1+d))</td>
<td>(v = 1/3)</td>
</tr>
<tr>
<td>(b = 1)</td>
<td>(u = 1/3(2+d))</td>
<td>(v = 0)</td>
</tr>
<tr>
<td>(d = 0)</td>
<td>(u = 1/3(1+b))</td>
<td>(v = 2/3 - u)</td>
</tr>
<tr>
<td>(d = 1)</td>
<td>(u = 1/3(2+b))</td>
<td>(v = 1 - u)</td>
</tr>
</tbody>
</table>

transformation \((b, d) \rightarrow (u, v)\) is a simple linear transformation, so the transformation
of the boundaries of the square in the \(bd\)-plane are sufficient to describe \(R_1\). From the
boundary calculations found in Table 3.3, we have that \(R_1\) is the quadrilateral with
vertices \((1, 0), (2/3, 0), (1/3, 1/3)\) and \((2/3, 1/3)\) in the \(uv\)-plane. Therefore, measures
corresponding to a vector \(q \in R_1\) can be modeled by a triangle measure of Type 1.
**Type 2 Triangle Measures** The triangle $t$ for Type 2 triangle measures is bounded by

\[ x_1 = x_2 - 1 \]
\[ x_1 = L_{cd}(x_2) = \frac{c - d}{c + 1} x_2 + d \]
\[ x_2 = 0. \]

Thus the moments for Type 2 triangle measures are

\[ M_{00}(\Delta^*, \mu) = \int_{\Delta^*} 1d\mu = \int_{t} Cdx = \int_{x_2 = \frac{c - d}{c + 1} x_2 + d}^{x_2 - 1} Cdx_1dx_2 \]
\[ = \frac{C}{2}(c + 1)(1 + d) \]  

(3.64)

\[ M_{10}(\Delta^*, \mu) = \int_{\Delta^*} x_1d\mu = \int_{t} Cx_1dx = \int_{x_2 = \frac{c - d}{c + 1} x_2 + d}^{x_2 - 1} Cx_1dx_1dx_2 \]
\[ = \frac{C}{6}(c + 1)(1 + d)(d + c - 1) \]  

(3.65)

\[ M_{01}(\Delta^*, \mu) = \int_{\Delta^*} x_2d\mu = \int_{t} Cx_2dx = \int_{x_2 = \frac{c - d}{c + 1} x_2 + d}^{x_2 - 1} Cx_2dx_1dx_2 \]
\[ = \frac{C}{6}(c + 1)^2(1 + d) \]  

(3.66)

(3.67)

Thus we define the transformation $(c, d) \rightarrow (u, v)$ by

\[ u(c, d) = \frac{d + c - 1}{3} \quad \text{and} \quad v(c, d) = \frac{c + 1}{3}. \]  

(3.68)

The inverse relation $(u, v) \rightarrow (c, d)$ is defined by

\[ c(u, v) = 3v - 1 \quad \text{and} \quad d(u, v) = 2 - 3v + 3u. \]  

(3.69)

Let $R_2$ denote the image of the square $-1 \leq c \leq 2$ and $0 \leq d \leq 1$ in the $uv$-plane. The transformation $(c, d) \rightarrow (u, v)$ is a simple linear transformation, so the transformation of the boundaries of the square in the $cd$-plane are sufficient to describe $R_2$. From the boundary calculations found in Table 3.4, we have that $R_2$ is the quadrilateral with vertices $(-2/3, 0), (-1/3, 0), (0, 1/3)$ and $(-1/3, 1/3)$ in the $uv$-plane. Therefore, measures corresponding to a vector $q \in R_2$ can be modeled by a triangle measure of Type 2.
Table 3.4 Type 2 Triangle Measures from the boundaries in $cd$-plane to the $uv$-plane

<table>
<thead>
<tr>
<th>$cd$-plane</th>
<th>Transformation</th>
<th>Image in $uv$-plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c = 0$</td>
<td>$u = 1/3(-1 + d)$</td>
<td>$v = 1/3$</td>
</tr>
<tr>
<td>$c = -1$</td>
<td>$u = 1/3(-2 + d)$</td>
<td>$v = 0$</td>
</tr>
<tr>
<td>$d = 0$</td>
<td>$u = 1/3(c - 1)$</td>
<td>$v = 1/3(1 + c)$</td>
</tr>
<tr>
<td>$d = 1$</td>
<td>$u = c/3$</td>
<td>$v = 1/3(1 + c)$</td>
</tr>
</tbody>
</table>

**Type 3 Triangle Measures**  Lastly, we consider Type 3 triangle measures. In this case, the triangle $t$ is bounded by

\[
\begin{align*}
    x_2 &= 1 - x_1 \\
    x_2 &= 1 + x_1 \\
    x_2 &= L_{bc}(x_1) = \frac{(c + b)(x_1 - b)}{c - b} + 1 - b.
\end{align*}
\]

Thus the moments for Type 3 triangle measures are

\[
\begin{align*}
    M_{00}(\Delta^*, \mu) &= \int_{c}^{0} \int_{L_{bc}(x_1)}^{1+x_1} Cdx_2dx_1 + \int_{0}^{b} \int_{L_{bc}(x_1)}^{1-x_1} Cdx_2dx_1 \\
    &= -Cbc \\
    M_{10}(\Delta^*, \mu) &= \int_{c}^{0} \int_{L_{bc}(x_1)}^{1+x_1} Cx_1dx_2dx_1 + \int_{0}^{b} \int_{L_{bc}(x_1)}^{1-x_1} Cx_1dx_2dx_1 \\
    &= -\frac{C}{3}bc(b + c) \\
    M_{01}(\Delta^*, \mu) &= \int_{c}^{0} \int_{L_{bc}(x_1)}^{1+x_1} Cx_2dx_2dx_1 + \int_{0}^{b} \int_{L_{bc}(x_1)}^{1-x_1} Cx_2dx_2dx_1 \\
    &= \frac{C}{3}bc(b - c - 3)
\end{align*}
\]

The transformation $(b, c) \rightarrow (u, v)$ is defined by

\[
\begin{align*}
    u(b, c) &= \frac{b + c}{3} \quad \text{and} \quad v(b, c) = \frac{-b + c + 3}{3}.
\end{align*}
\]

Then we have the inverse relation $(u, v) \rightarrow (c, d)$ which is

\[
\begin{align*}
    b(u, v) &= 3/2(1 - v + u) \quad \text{and} \quad c(u, v) = 3/2(-1 + v + u).
\end{align*}
\]

From the transformation of the boundaries in the $bc$-plane, we describe the im-
Table 3.5  Type 3 Triangle Measures from $bc$-plane boundaries to the $uv$-plane

<table>
<thead>
<tr>
<th>$bc$-plane</th>
<th>Transformation</th>
<th>Image in $uv$-plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b = 0$</td>
<td>$u = c/3$</td>
<td>$v = 1 + c/3$</td>
</tr>
<tr>
<td>$b = 1$</td>
<td>$u = 1/3(c + 1)$</td>
<td>$v = 1/3(2 + c)$</td>
</tr>
<tr>
<td>$c = 0$</td>
<td>$u = b/3$</td>
<td>$v = 1 - b/3$</td>
</tr>
<tr>
<td>$c = -1$</td>
<td>$u = 1/3(b - 1)$</td>
<td>$v = 1/3(2 - b)$</td>
</tr>
</tbody>
</table>

Figure 3.24  The $uv$-plane colored according to the types. Type 1 is blue and the symmetric case is light blue. Type 2 is purple and the symmetric case is light purple. Type 3 is yellow.

For each type of triangle measure, equations have been presented for the calculation of the parameters $b, c, \text{ and } d$ given $q = (u, v)$ of the measure over the parent. For each type of triangle measure, equations have been presented for the calculation of the parameters $b, c, \text{ and } d$ given $q = (u, v)$ of the measure over the parent. These parameters determine the triangle $t$. In the case of intervals, the constant $C$ was simply the zeroth moment of the parent. In the case of triangles, it is not that simple.
Lemma 3.6. Let \( \rho \) be any measure over \( \Delta^* \) with \( q(\rho) \) in \( R_1 \cup R_2 \cup R_3 \) (or in the reflection of \( R_1 \cup R_2 \cup R_3 \) across the v-axis). Then there is a unique \( \mu \) of the form \( d\mu = C\chi_t dx \) with \( t \in \mathcal{D} \) such that \( Q_X(\Delta^*, \mu) = Q_X(\Delta^*, \rho) \).

Proof. First identify in which region \( q(\rho) = (u(\rho), v(\rho)) \) lies. If \( q(\rho) \in R_i \) for \( i \in \{1, 2, 3\} \), then the representative measure is of Type \( i \).

For \( i = 1 \), using the inverse relationship Eq (3.62) let
\[
b = 1 - 3v(\rho) \quad \text{and} \quad d = -2 + 3v(\rho) + 3u(\rho). \tag{3.76}
\]

Then let
\[
C_1 = \frac{2M_{00}(\Delta^*, \rho)}{(d - 1)(b - 1)}. \tag{3.77}
\]

Define \( t_{bd} \) to be the triangle with vertices \((b, 1 - b), (d, 0), \) and \((1, 0)\) and \( \mu \) to be the measure such that \( d\mu = C_1\chi_{t_{bd}} dx \). Equations (3.57), (3.58), and (3.59) then give us that
\[
M_{00}(\Delta^*, \mu) = \frac{C_1}{2}(d - 1)(b - 1) = \frac{M_{00}(\Delta^*, \rho)}{(d - 1)(b - 1)}(d - 1)(b - 1) = M_{00}(\Delta^*, \rho)
\]
\[
M_{10}(\Delta^*, \mu) = \frac{C_1}{6}(d - 1)(b - 1)(1 + b + d) = \frac{M_{00}(\Delta^*, \rho)}{3}(1 + b + d) = \frac{M_{00}(\Delta^*, \rho)}{3}(1 - 3v(\rho) - 2 + 3v(\rho) + 3u(\rho)) = M_{00}(\Delta^*, \rho)u(\rho) = M_{10}(\Delta^*, \rho)
\]
\[
M_{01}(\Delta^*, \mu) = -\frac{C_1}{6}(d - 1)(b - 1)^2 = \frac{-M_{00}(\Delta^*, \rho)}{3}(b - 1) = M_{00}(\Delta^*, \rho)v(\rho) = M_{01}(\Delta^*, \rho).
\]

Thus \( Q_X(\Delta^*, \mu) = Q_X(\Delta^*, \rho) \).

For \( i = 2 \), the approach is similar. Let
\[
c = 3v(\rho) - 1 \quad \text{and} \quad d = 2 - 3v(\rho) + 3u(\rho), \tag{3.78}
\]
as in equation (3.69). Then let

$$C_2 = \frac{2M_{00}(\Delta^*, \rho)}{(d + 1)(c + 1)}. \quad (3.79)$$

Define $t_{cd}$ to be the triangle with vertices $(c, 1 + c), (d, 0)$, and $(-1, 0)$ and $\mu$ to be the measure such that $d\mu = C_2 \chi_{t_{cd}} d\mathbf{x}$. Equations (3.64), (3.65), and (3.66) then give us that

$$M_{00}(\Delta^*, \mu) = \frac{C_2}{2} (d + 1)(c + 1) = M_{00}(\Delta^*, \rho)$$

$$M_{10}(\Delta^*, \mu) = \frac{C_2}{6} (d + 1)(c + 1)(d + c - 1) = \frac{M_{00}(\Delta^*, \rho)}{3}(d + c - 1)$$

$$= \frac{M_{00}(\Delta^*, \rho)}{3}(2 - 3v(\rho) + 3u(\rho) + 3v(\rho) - 1 - 1)$$

$$= M_{00}(\Delta^*, \rho)u(\rho) = M_{10}(\Delta^*, \rho)$$

$$M_{01}(\Delta^*, \mu) = \frac{C_2}{6} (d + 1)(c + 1)^2 = \frac{M_{00}(\Delta^*, \rho)}{3}(c + 1)$$

$$= M_{00}(\Delta^*, \rho)v(\rho) = M_{01}(\Delta^*, \rho).$$

Thus $Q_X(\Delta^*, \mu) = Q_X(\Delta^*, \rho)$.

For $i = 3$, let

$$b = 3/2(1 - v(\rho) + u(\rho)) \quad \text{and} \quad c = 3/2(-1 + v(\rho) + u(\rho)), \quad (3.80)$$

as in equation (3.75). Then let

$$C_3 = \frac{-M_{00}(\Delta^*, \rho)}{bc}. \quad (3.81)$$

Define $t_{bc}$ to be the triangle with vertices $(b, 1 - b), (c, 1 + c)$, and $(0, 1)$ and $\mu$ to be the measure such that $d\mu = C_3 \chi_{t_{bc}} d\mathbf{x}$. Equations (3.71), (3.72), and (3.73) then give
us that

\[
M_{00}(\Delta^*, \mu) = -C_3 bc = M_{00}(\Delta^*, \rho)
\]

\[
M_{10}(\Delta^*, \mu) = \frac{-C_3}{3} bc (b + c) = \frac{M_{00}(\Delta^*, \rho)}{3} (b + c)
\]

\[
= \frac{M_{00}(\Delta^*, \rho)}{3} (3/2(1 - v(\rho) + u(\rho)) + 3/2(-1 + v(\rho) + u(\rho)))
\]

\[
= M_{00}(\Delta^*, \rho) u(\rho) = M_{10}(\Delta^*, \rho)
\]

\[
M_{01}(\Delta^*, \mu) = \frac{C_3}{3} bc (-b + c + 3) = \frac{-M_{00}(\Delta^*, \rho)}{3} (-b + c + 3)
\]

\[
= \frac{-M_{00}(\Delta^*, \rho)}{3} (-3/2(1 - v(\rho) + u(\rho)) + 3/2(-1 + v(\rho) + u(\rho)) + 3)
\]

\[
= M_{00}(\Delta^*, \rho) v(\rho) = M_{01}(\Delta^*, \rho).
\]

Thus \( Q_X(\Delta^*, \mu) = Q_X(\Delta^*, \rho) \).

It remains to find \( \mu \) for \( q(\rho) \) in the reflection of \( R_1 \cup R_2 \cup R_3 \) across the \( v \)-axis. Suppose \( q(\rho) \) is in the reflection of some \( R_i \) for \( i \in \{1, 2, 3\} \). Then we replace \( q(\rho) = (u, v) \) with \( q'(\rho) = (-u, v) \). From the above work, we calculate the measure \( \mu' \) of the form \( d\mu' = C\chi_t d\mathbf{x} \) with \( q(\mu') = q'(\rho) \). Let \( \mu \) be the measure such that \( d\mu = C\chi_t d(-x_1)dx_2 \). Define \( t' \) as the triangle \( t \) reflected across the \( x_2 \)-axis. Then \( d\mu = C\chi_{t'} d\mathbf{x} \). Thus \( Q_X(\Delta^*, \mu) = Q_X(\Delta^*, \rho) \).

\[ \square \]

**Complementary Measures**

We turn our attention now to the complementary measures \( \mu^C \) of the form \( d\mu^C = C(1 - \chi_t)d\mathbf{x} \) (Figures 3.25, 3.26, 3.27). We start by examining the transformations to the \( uv \)-plane for each type. The complementary measures do not have the simple linear relationship with the \( uv \)-plane exhibited by the triangle measures. Thus we first show for each type that there exists a unique transformation from the \( uv \)-plane to the parameters \( b, c, \) and \( d \). Later we show the calculation of this transformation.

For the computation of the moments for complementary measures, we us the
Lebesgue measure over the entire triangle $\Delta^*$. Note that

$$\int_{\Delta^*} 1dx = \int_0^1 \int_{x_2-1}^{1-x_2} 1dx_1dx_2 = 1$$ (3.82)  
$$\int_{\Delta^*} x_1dx = \int_0^1 \int_{x_2-1}^{1-x_2} x_1dx_1dx_2 = 0$$ (3.83)  
$$\int_{\Delta^*} x_2dx = \int_0^1 \int_{x_2-1}^{1-x_2} x_2dx_1dx_2 = 1/3$$ (3.84)

and Lebesgue measure over the entire triangle $\Delta^*$ corresponds to the $uv$-vector $(0, 1/3)$. In future considerations we need not compute the transformations for $(0, 1/3)$.

**Type 1 Complementary Measures**  The moments in terms of $b, d,$ and $C$ for the complementary measures of Type 1 are

$$M_{00}(\Delta^*, \mu^C) = \int_{\Delta^*} 1d\mu^C = \int_{\Delta^*} Cdx - \int_t Cdx$$  
$$= C - C/2(d - 1)(b - 1)$$ (3.85)  
$$M_{10}(\Delta^*, \mu^C) = \int_{\Delta^*} x_1d\mu^C = \int_{\Delta^*} Cx_1dx - \int_t Cx_1dx$$  
$$= -C/6(d - 1)(b - 1)(1 + b + d)$$ (3.86)  
$$M_{01}(\Delta^*, \mu^C) = \int_{\Delta^*} x_2d\mu^C = \int_{\Delta^*} Cx_2dx - \int_t Cx_2dx$$  
$$= C/3 + C/6(d - 1)(b - 1)^2$$ (3.87)

Let $M^C_1$ be the transformation $(b, d) \rightarrow (u, v)$ given by

$$u(b, d) = \frac{M_{10}(\Delta^*, \mu^C)}{M_{00}(\Delta^*, \mu^C)} = \frac{(d - 1)(b - 1)(1 + b + d)}{-6 + 3(d - 1)(b - 1)}$$ (3.88)
and
\[
v(b, d) = \frac{M_{01}(\Delta^*, \mu^c)}{M_{00}(\Delta^*, \mu^c)} = \frac{(d - 1)(b - 1)^2 + 2}{6 - 3(d - 1)(b - 1)} \tag{3.89}
\]

\(M^c_1\) is continuous for all \(b\) and \(d\) in \([0, 1]\). Thus examination of the transformation of the boundaries is sufficient to determine the region of the image. \(M^c_1\) of the boundaries of the square \(0 \leq b \leq 1\) and \(0 \leq d \leq 1\) from the \(bd\)-plane to the \(uv\)-plane are given in Table 3.7. Hence, the image is bounded by the curves \(v = \frac{2 - 9u^2}{3(2 + 3u)}\) and \(v = 1/3\).

<table>
<thead>
<tr>
<th>(bd)-plane</th>
<th>Transformation</th>
<th>Image in (uv)-plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b = 0)</td>
<td>(u = 1/3(d - 1)) (v = 1/3)</td>
<td>(v = 1/3) ((0, 1/3))</td>
</tr>
<tr>
<td>(b = 1)</td>
<td>(u = 0) (v = 1/3)</td>
<td>(v = 2 - 9u^2) (3(2 + 3u))</td>
</tr>
<tr>
<td>(d = 0)</td>
<td>(u = 1/3(b - 1)) (v = \frac{1 + 2b - b^2}{3(b + 1)})</td>
<td>(v = 1/3) ((0, 1/3))</td>
</tr>
<tr>
<td>(d = 1)</td>
<td>(u = 0) (v = 1/3)</td>
<td></td>
</tr>
</tbody>
</table>

Note that the \(uv\)-vector \((0, 1/3)\) corresponds to Lebesgue measure over the entire triangle \(\Delta^*\). Define \(S_1\) to be the set \(\{(b, d) : b, d \in [0, 1]\}\), and let \(T_1\) be the region in the \(uv\)-plane bounded by the curves \(v = \frac{2 - 9u^2}{3(2 + 3u)}\) and \(v = 1/3\) with a hole at \((0, 1/3)\).

**Lemma 3.7.** The map \(M^c_1 : S_1 \to T_1\) is invertible.

**Proof.** Note that \(M^c_1(S_1) \subseteq T_1\) from the boundary analysis above. The Jacobian of \(M^c_1\) is
\[
J = \frac{(d - 1)(b - 1)[(d - 1)^2b^2 - 2d^2b - 2db + (d + 1)^2]}{9(2 - (d - 1)(b - 1))^3}
\]
We wish to show that \(J \neq 0\) for all \((b, d) \in S_1\). The factors \((d - 1)\) and \((b - 1)\) are disregarded as \(b \neq 1\) and \(d \neq 1\). Dissecting and regrouping the terms of the numerator, we have that
\[
(d - 1)^2b^2 + (1 - d^2b) + (d^2 - d^2b) + (2d - 2db).
\]
Each term is greater than or equal to zero as \( b, d \in [0, 1) \). Furthermore, the term \((1 - d^2 b)\) is strictly greater than zero because \( b, d \in [0, 1) \). Thus the Jacobian of \( M^C_1 \) is nonzero on \( S_1 \). Hence we have that \( M^C_1 \) is invertible.

**Type 2 Complementary Measures**  We start by defining the map \( M^C_2 \). Using the relationship between the complementary measures and the triangle measures, we have that

\[
M_{00}(\Delta^*, \mu^C) = C - C/2(d + 1)(c + 1) \quad (3.90)
\]
\[
M_{10}(\Delta^*, \mu^C) = -C/6(d + 1)(c + 1)(c + d - 1) \quad (3.91)
\]
\[
M_{01}(\Delta^*, \mu^C) = C/3 - C/6(d + 1)(c + 1)^2. \quad (3.92)
\]

Therefore, the map \( M^C_2 \) is given by

\[
u = \frac{(d + 1)(c + 1)(c + d - 1)}{3(d + 1)(c + 1) - 6} \quad \text{and} \quad v = \frac{(d + 1)(c + 1)^2 - 2}{3(d + 1)(c + 1) - 6}. \quad (3.93)
\]

\( M^C_2 \) is continuous for \((c, d) \neq (0, 1)\). Thus examination of the transformation of the boundaries is sufficient to determine the region of the image. \( M^C_2 \) of the boundaries of the square \(-1 \leq c \leq 0\) and \(0 \leq d \leq 1\) from the \(cd\)-plane to the \(uv\)-plane are given in Table 3.8. We have the additional boundary \( v = 1 - u \), since \( u \) corresponds to the average \( x_1 \) and \( v \) corresponds to the average \( x_2 \). This boundary also appears as a limit when both \( c \to 0 \) and \( d \to 1 \) and is discussed in Section 3.3. Hence, the image is bounded by the curves \( v = \frac{9u^2 - 2}{3(3u - 2)} \), \( v = 1/3 \), \( v = u + 1/3 \), and \( v = 1 - u \). Let

<table>
<thead>
<tr>
<th>( cd )-plane</th>
<th>Transformation</th>
<th>Image in ( uv )-plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c = 0 )</td>
<td>( u = 1/3(1 + d) )</td>
<td>( v = 1/3 )</td>
</tr>
<tr>
<td>( c = -1 )</td>
<td>( u = 0 )</td>
<td>( v = 1/3 )</td>
</tr>
<tr>
<td>( d = 0 )</td>
<td>( u = 1/3(1 + c) )</td>
<td>( v = \frac{-1 + 2c + c^2}{3(c - 1)} )</td>
</tr>
<tr>
<td>( d = 1 )</td>
<td>( u = 1/3(1 + c) )</td>
<td>( v = 1/3(2 + c) )</td>
</tr>
</tbody>
</table>

average \( x_1 \) and \( v \) corresponds to the average \( x_2 \). This boundary also appears as a limit when both \( c \to 0 \) and \( d \to 1 \) and is discussed in Section 3.3. Hence, the image is bounded by the curves \( v = \frac{9u^2 - 2}{3(3u - 2)} \), \( v = 1/3 \), \( v = u + 1/3 \), and \( v = 1 - u \). Let
\( S_2 = \{(c, d) \in (-1, 0] \times [0, 1] \setminus (0, 1)\}, \) and let \( T_2 \) be the region in the \( uv \)-plane with 
\[ v \geq \frac{9u^2 - 2}{3(3u - 2)}, \quad v \geq 1/3, \quad v \leq u + 1/3, \quad v < 1 - u, \] and a hole at \((0, 1/3)\).

**Lemma 3.8.** The map \( M_2^C : S_2 \to T_2 \) is invertible.

**Proof.** The Jacobian of \( M_2^C \) is
\[
J = \frac{(d + 1)(c + 1)[(d + 1)^2c^2 + 2d^2c + 2dc + (d - 1)^2]}{9(2 - (d + 1)(c + 1))^3}
\]
We show that the Jacobian is nonzero for all \((c, d) \in S_2\). The factors \((d + 1)\) and \((c + 1)\) are disregarded as \(c \neq -1\) and \(d \neq -1\). Next we examine the rest of the numerator, which is
\[(d + 1)^2c^2 + 2d(d - 1)c + (d - 1)^2.\]
Each term is greater than or equal to 0 since \(c \in (-1, 0]\) and \(d \in [0, 1]\). If \(c \neq 0\), then the term \((d+1)^2c^2 > 0\). If \(d \neq 1\), then the term \((d - 1)^2 > 0\). Note that \((0, 1) \notin S_2\), so the numerator of \(J\) is strictly greater than zero. Thus, the Jacobian of \( M_2^C \) is nonzero on \(S_2\). So we have that \( M_2^\ell \) is invertible. \(\square\)

Now we repeat the process above and define the map \( M_3^C \). We have that
\[
M_{00}(\Delta^*, \mu^C) = C + Cbc \quad (3.94)
\]
\[
M_{10}(\Delta^*, \mu^C) = C/3bc(c + d) \quad (3.95)
\]
\[
M_{01}(\Delta^*, \mu^C) = C/3 - C/3bc(b - c - 3) \quad (3.96)
\]
Then the map \( M_3^C \) is given by
\[
u = \frac{bc(c + b)}{3(1 + bc)} \quad \text{and} \quad v = \frac{-bc(b - c - 3) + 1}{3(1 + bc)} \quad (3.97)
\]
The map \( M_3^C \) is continuous for all \((b, c) \in [0, 1] \times [-1, 0]\setminus(1, -1)\). Table 3.9 contains the boundary calculations. We have the additional boundary \(v = 0\), since \(u\) corresponds to the average \(x_1\) and \(v\) corresponds to the average \(x_2\). This boundary also appears as a limit when both \(b \to 1\) and \(c \to -1\) and is discussed in Section 3.3.
Table 3.9  Type 3 Complementary Measures from $bc$ boundaries to $uv$

<table>
<thead>
<tr>
<th>$bc$-plane</th>
<th>Transformation</th>
<th>Image in $uv$-plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b = 0$</td>
<td>$u = 0$</td>
<td>$v = 1/3$</td>
</tr>
<tr>
<td>$b = 1$</td>
<td>$u = c/3$</td>
<td>$v = 1/3(1 + c)$</td>
</tr>
<tr>
<td>$c = 0$</td>
<td>$u = 0$</td>
<td>$v = 1/3$</td>
</tr>
<tr>
<td>$c = -1$</td>
<td>$u = b/3$</td>
<td>$v = 1/3(1 - b)$</td>
</tr>
</tbody>
</table>

Hence, the image is bounded by the curves $v = 1/3 - u$, $v = 1/3 + u$, and $v = 0$. Let $S_3 = \{(b, c) \in (0, 1] \times [-1, 0) \setminus (1, -1)\}$, and let $T_3$ be the region in the $uv$-plane with $v \leq 1/3 - u$, $v \leq 1/3 + u$, $v > 0$, and a hole at $(0, 1/3)$.

**Lemma 3.9.** The map $M^C_3 : S_3 \rightarrow T_3$ is invertible.

**Proof.** The Jacobian of $M^C_3$ is

$$J = \frac{2bc(b^2c^2 + 3bc + b - c)}{9(1 + bc)^3}$$

Next we show that the Jacobian is nonzero for all $(b, c) \in S_3$. The factor $2bc$ is disregarded as $b \neq 0$ and $c \neq 0$. Then the numerator is rewritten with the substitution $c = -c$ so that now $c \in (0, 1]$ instead of in $[-1, 0)$, and denote it by $N$

$$N = b^2c^2 - 3bc + b + c, \quad (3.98)$$

and $N \neq 0$ if and only if $J \neq 0$. Then $b$ and $c$ nonzero allows us to divide by $bc$. So

$$\frac{N}{bc} = bc + \frac{1}{c} + \frac{1}{b} - 3.$$

The terms $bc$, $1/c$, and $1/b$ are all nonnegative so we apply the inequality of the arithmetic and the geometric mean. We use the strictly greater than form since we do not have that all the terms are equal. This gives us that

$$\frac{N}{bc} > 3 \sqrt[3]{bc \cdot 1/c \cdot 1/b} - 3 = 3 - 3 = 0.$$

Thus the Jacobian of $M^C_3$ is nonzero on $S_3$, and we have that $M^C_3$ is invertible. \qed
Figure 3.28  The $uv$-plane colored according to the triangle and complementary measure types. $R_i$ are triangle measures and $T_i$ are complementary measures for $i = 1, 2, 3$.

**Limiting Cases on the Boundary**

Lemmas 3.7, 3.8, 3.9 show that the relations are invertible for the vector $q$ in $\Delta^*$ except for $(0, 1/3)$ and some of the boundaries. The $uv$-vector $(0, 1/3)$ is associated with the Lebesgue measure over $\Delta^*$, so no further analysis is needed for this vector. There are three sections of the boundary not included in the regions $R_i, T_j$, and their symmetric counterparts:

1. $v = 1 - u$ for $u \in (1/3, 2/3)$

2. $v = 1 + u$ for $u \in (-2/3, 1/3)$

3. $v = 0$ for $u \in [-1/3, 1/3]$.

These boundary sections correspond to limiting cases which appear when the area of the subdomain defining the complementary measure ($\Delta^* t$) is arbitrarily small. In these situations, the measure is concentrated along an edge. The first arises as a limiting case for Type 2 complementary measures. The second arises for the symmetric
case for Type 2 complementary measures. Lastly, the third arises for a limiting case for Type 3 complementary measures.

The Type 2 complementary measures limiting case occurs when $c = 0$ and $d = 1$. Notice the ‘and’ statement here. This situation is similar to the limiting case we managed with the intervals. Depending on the way $(c, d)$ approaches $(0, 1)$, we will have different $uv$-coordinates. Start by considering $c = 0$ and $d$ approaching 1. Then

$$u = 1/3(1 + d) \to 2/3 \quad \text{and} \quad v = 1/3.$$

On the other hand when $d = 1$ and $c$ approaches 0, then

$$u = 1/3(1 + c) \to 1/3 \quad \text{and} \quad v = 1/3(2 + c) \to 2/3.$$

Let $\tau > 0$ and $\epsilon > 0$, consider $c = \epsilon$ and $d = 1 - \tau \epsilon$. As $\epsilon \to 0$ then $c \to 0$ and $d \to 1$. We rewrite $u$ and $v$ in terms of $\tau$ and $\epsilon$. Then $\epsilon \to 0$ implies that $u$ approaches $2(\tau - 1) / 3(\tau - 2)$ and $v$ approaches $\tau - 4 / 3(\tau - 2)$. Note that $v = 1 - u$ and $u$ can be any value in $(1/3, 2/3)$ since

$$\lim_{\tau \to 0} u = 1/3 \quad \text{and} \quad \lim_{\tau \to \infty} u = 2/3.$$

Thus this limiting case covers $v = 1 - u$ for $u \in [1/3, 2/3]$ in the $uv$-plane. In this case, the measure is concentrated on the edge $v = 1 - u$. Therefore, we consider the left child $\Delta_1^*$ to be empty and the quantities of the right child $Q_X(\Delta_2^*) = Q_X(\Delta^*)$. The corresponding symmetric case covers $v = 1 + u$ for $u \in [-2/3, 1/3]$ and we consider the quantities of the left child $Q_X(\Delta_1^*) = Q_X(\Delta^*)$ and the right child $\Delta_2^*$ to be empty.

The limiting case in complementary measures of Type 3 occur when $b = 1$ and $c = -1$. First consider $c = -1$ and $b$ approaching 1. Then

$$u = b/3 \to 1/3 \quad \text{and} \quad v = 1/3(1 - b) \to 0.$$

On the other hand when $b = 1$ and $c$ approaches 0, then

$$u = c/3 \to -1/3 \quad \text{and} \quad v = 1/3(1 + c) \to 0.$$
Now for $\tau, \epsilon > 0$, consider $b = 1 - \epsilon$ and $c = -1 + \tau \epsilon$. As $\epsilon$ approaches 0, $u$ approaches $\frac{\tau - 1}{-3(\tau + 1)}$ and $v \to 0$. Notice that $u$ can be any value in $(-1/3, 1/3)$ since

$$\lim_{\tau \to 0} u = 1/3 \quad \text{and} \quad \lim_{\tau \to \infty} u = -1/3.$$ 

Hence this limiting case covers $v = 0$ for $u \in [-1/3, 1/3]$ on the $uv$-plane. In this case, the measure is concentrated on the edge $v = 0$. If $u > 0$, then we consider $\Delta_1^*$ to be empty and $Q_X(\Delta_2^*) = Q_X(\Delta^*)$. If $u < 0$, then we consider $\Delta_2^*$ to be empty and $Q_X(\Delta_1^*) = Q_X(\Delta^*)$. If $u = 0$, then we let $Q_X(\Delta_1^*) = \frac{1}{2}Q_X(\Delta^*)$ and $Q_X(\Delta_2^*) = \frac{1}{2}Q_X(\Delta^*)$.

**Computation of Parameters for Complementary Measures**

Given the quantities $Q_X(\Delta^*, \rho)$ for some measure $\rho$ we wish to find a representative measure $\mu$ such that $Q_X(\Delta^*, \mu) = Q_X(\Delta^*, \rho)$. In Lemma 3.6 we have given a method for determining the parameters for triangle measures ($q(\rho) \in R_1 \cup R_2 \cup R_3$ or symmetric). Now we develop the method for complementary measures where $q(\rho) \in T_1 \cup T_2 \cup T_3$ or symmetric, in which case the representative measure is of the form $\mu^C$ with $d\mu^C = C(1 - \chi_t)dx$. The idea is to write the parameters $b, c, d$ in terms of $C$ and then use them to solve for $C$. This requires solving a quadratic equation. Once we know the parameters and $C$, then the representative measure is fully known and the children’s *distribution-dependent* quantities can easily be calculated. We will need to look at each type individually.

For Type 1 complementary measures we have the following system of equations that we need to solve for $(b, d, C)$ given the moments of $t^C$.

$$\frac{C}{2}(d - 1)(b - 1) = C - M_{00}(\Delta^*, \rho^C) \quad (3.99)$$
$$\frac{C}{6}(d - 1)(b - 1)(1 + b + d) = -M_{10}(\Delta^*, \rho^C) \quad (3.100)$$
$$-\frac{C}{6}(d - 1)(b - 1)^2 = C/3 - M_{01}(\Delta^*, \rho^C) \quad (3.101)$$
Dividing (3.101) by (3.99), we have that
\[ -(b - 1)/3 = \frac{C/3 - M_{01}(\Delta^*, \rho^\ell)}{C - M_{00}(\Delta^*, \rho^\ell)} \] (3.102)
\[ \Rightarrow b = \frac{-C + 3M_{01}(\Delta^*, \rho^\ell)}{C - M_{00}(\Delta^*, \rho^\ell)} + 1 \]

Dividing (3.100) by (3.99) and substituting (3.104), we have that
\[ 1/3(1 + b + d) = \frac{-M_{10}(\Delta^*, \rho^\ell)}{C - M_{00}(\Delta^*, \rho^\ell)} \] (3.103)
\[ \Rightarrow d = \frac{-3M_{10}(\Delta^*, \rho^\ell) + C - 3M_{01}(\Delta^*, \rho^\ell)}{C - M_{00}(\Delta^*, \rho^\ell)} - 2 \]

Now we plug \( b \) and \( d \) back into (3.99) and solve for \( C \), which requires solving the quadratic function
\[ 3C^2 (M_{10} - M_{01} + M_{00}) + 3C \left( -3M_{10}M_{01} - 3M_{01}^2 + 3M_{00}M_{01} - 2M_{00}^2 \right) + 2M_{00}^3 = 0 \]

We need to be sure that \( M_{10} + M_{00} - M_{01} \) is not zero, which occurs when \( v = u + 1 \). Fortunately, the complementary Type 1 is nowhere near satisfying that equation. We reject the solution for \( C \) which implies that \( b \) or \( d \) is outside of their ranges.

Using the same strategy as in Type 1, we have for complementary measures of Type 2 that
\[ 1/3(c + 1) = \frac{C/3 - M_{01}(\Delta^*, \rho^\ell)}{C - M_{00}(\Delta^*, \rho^\ell)} \] (3.104)
\[ \Rightarrow c = \frac{C - 3M_{01}(\Delta^*, \rho^\ell)}{C - M_{00}(\Delta^*, \rho^\ell)} - 1 \]

Dividing (3.100) by (3.99) and substituting (3.104), we have that
\[ 1/3(c + d - 1) = \frac{-M_{10}(\Delta^*, \rho^\ell)}{C - M_{00}(\Delta^*, \rho^\ell)} \] (3.105)
\[ \Rightarrow d = \frac{-3M_{10}(\Delta^*, \rho^\ell) - C + 3M_{01}(\Delta^*, \rho^\ell)}{C - M_{00}(\Delta^*, \rho^\ell)} + 2 \]

Now we plug \( c \) and \( d \) and solve for \( C \), which requires solving the quadratic function
\[ 3C^2 ( -M_{10} - M_{01} + M_{00}) + 3C \left( 3M_{10}M_{01} - 3M_{01}^2 + 3M_{00}M_{01} - 2M_{00}^2 \right) + 2M_{00}^3 = 0 \]
We need to be sure that $-M_{10} + M_{00} - M_{01}$ is not zero, which occurs when $v = 1 - u$. This only occurs in the limiting cases which we have already discussed in Section 3.3. We reject the solution for $C$ which implies that $c$ or $d$ is outside of their ranges.

Again we use the same strategy as in Type 1.

\[-1/3(b - c - 3) = \frac{C/3 - M_{01}(\Delta^*, \rho^E)}{C - M_{00}(\Delta^*, \rho^E)} \]  
\[\Rightarrow b - c = \frac{-C + 3M_{01}(\Delta^*, \rho^E)}{C - M_{00}(\Delta^*, \rho^E)} + 3 \]  
\[\text{(3.106)}\]

We also have that

\[b + c = \frac{-3M_{10}(\Delta^*, \rho^E)}{C - M_{00}(\Delta^*, \rho^E)} \]  
\[\text{(3.107)}\]

Adding the equations (3.106) and (3.107) we have that

\[\Rightarrow b = \frac{-3M_{10}(\Delta^*, \rho^E) - C + 3M_{01}(\Delta^*, \rho^E)}{2C - 2M_{00}(\Delta^*, \rho^E)} + 3/2 \]  
\[\text{(3.108)}\]

Then

\[\Rightarrow c = \frac{-3M_{10}(\Delta^*, \rho^E) + C - 3M_{01}(\Delta^*, \rho^E)}{2C - 2M_{00}(\Delta^*, \rho^E)} - 3/2 . \]  
\[\text{(3.109)}\]

Now we plug in $b$ and $c$ and solve for $C$, which requires solving the quadratic function

\[12C^2M_{01} + C \left(-9M_{10}^2 + 9M_{01}^2 - 18M_{00}M_{01} - 3M_{00}^2\right) + 4M_{00}^3 = 0 \]

We need to be sure that $M_{01}$ is not zero, which occurs when $v = 0$. The complementary measures of Type 3 have a limiting case there which we previously discussed in Section 3.3. We reject the solution for $C$ which implies that $b$ or $c$ is outside of their ranges.

**Blending**

Let $\rho$ be a measure with quantities $Q_X([-1, 1], \rho)$. Then we use the methods above to find an approximation for the children using either the weight or subdomain measure.
Figure 3.29 The uv-plane highlighting the overlap of the weighted measures and the subdomain measures.

schemes. The downside to the weight scheme is that there are measures that give quantities not possible for the weighted scheme. That is to say triangle

\[ \Delta_w = ((0, 1/2), (-1/4, 1/4), (1/4, 1/4)) \]

does not cover the whole triangle \(((0, 1), (-1, 0), (1, 0))\). On the other hand, all measures can be represented by a subdomain measure. When a measure is close to but not exactly Lebesgue, the representative measure assumes there is a gap in the measure. Similar to the interval case, this is not good because our point cloud applications will not have perfectly Lebesgue measure even when the Lebesgue measure is the most appropriate. Therefore, we choose to blend two schemes. Let \( Q_X(\Delta^*_1, \eta) \) and \( Q_X(\Delta^*_2, \eta) \) be the quantities calculated from the weighted scheme and \( Q_X(\Delta^*_1, \mu) \) and \( Q_X(\Delta^*_2, \mu) \) be the quantities from the subdomain scheme. There is no weighted approximation whenever \( q(\rho) \) lies outside of \( \Delta_w \). In this case, we use only the subdomain measure results.

Now assume that \( q(\rho) \in \Delta_w \). Let \((\lambda_0, \lambda_1, \lambda_2)\) be the barycentric coordinates of \( q(\rho) \) with respect to the triangle \( \Delta_w \), and denote the minimum by \( \lambda_m = \min\{\lambda_0, \lambda_1, \lambda_2\} \). When \( q(\rho) \) is on the boundary of \( \Delta_w \), the smallest barycentric coordinate would be zero \((\lambda_m = 0)\). When \( q(\rho) = (0, 1/3) \), the vector corresponding to the Lebesgue measure, we have that all of the barycentric coordinates are equal to \( 1/3 \), so \( \lambda_m = 1/3 \).
Therefore, we approximate the real $Q_X$ with $\tilde{Q}_X$ given by

$$\tilde{Q}_X(\Delta_i^*) = 3\lambda_m Q_X(\Delta_i^*, \eta) + (1 - 3\lambda_m) Q_X(\Delta_i^*, \mu) \tag{3.110}$$

for $i = 1, 2$.

**Approximation of the Value-Dependent Quantities**

Now that we have the *distribution-dependent* quantities $Q_X(\Delta_1^*, d\rho_X)$ and $Q_X(\Delta_2^*, d\rho_X)$, it remains to find $Q_Y(\Delta_1^*, d\rho_X)$ and $Q_Y(\Delta_2^*, d\rho_X)$. The *value-dependent* quantity is not as local as the *distribution-dependent* quantities. Therefore we include the quantities of the three neighbor triangles $\Delta_B$, $\Delta_L$, and $\Delta_R$ in the calculations. Here we use a fairly straightforward method to calculate the *value-dependent* quantities of the children as a linear combination of *value-dependent* quantities of the parent and parent’s neighbors. Let $\Delta_B$, $\Delta_L$, and $\Delta_R$ be the neighbors of $\Delta^*$ of the same level. More specifically, $\Delta_B$ shares the edge $(-1,0)$ and $(1,0)$, $\Delta_L$ shares the edge $(-1,0)$ and $(0,1)$, and $\Delta_R$ shares the edge $(0,1)$ and $(1,0)$. If desired, one could substitute another method for solving for $Q_Y(\Delta_1^*, d\rho_X)$ and $Q_Y(\Delta_2^*, d\rho_X)$ as long as it depends only upon the *distribution-dependent* quantities of the parent ($\Delta^*$), the parent’s neighbors ($\Delta_B, \Delta_L, \Delta_R$), and the children ($\Delta_1^*, \Delta_2^*$).

Then

$$Q_Y(\Delta_1^*) = c_0 Q_Y(\Delta_B) + c_1 Q_Y(\Delta_L) + c_2 Q_Y(\Delta_R) + c_3 Q_Y(\Delta^*) \tag{3.111}$$

$$Q_Y(\Delta_2^*) = d_0 Q_Y(\Delta_B) + d_1 Q_Y(\Delta_L) + d_2 Q_Y(\Delta_R) + d_3 Q_Y(\Delta^*) \tag{3.112}$$

for some coefficients $c_0, c_1, c_2, c_3, d_0, d_1, d_2$ and $d_3$. We also desire the approximation to preserve the polynomial $y$-values. Hence, we use the test functions $1, x_1$ and $x_2$ to calculate $Q_Y(\Delta_1^*)$ and $Q_Y(\Delta_2^*)$. That leaves an additional degree of freedom, so we add the requirement that the weights for the left and right triangles are symmetric.

The idea is that the weights should be able to balance. Equations (3.111), (3.112),
and the test functions give us the following system of equations:

\[
\begin{bmatrix}
\int_{\Delta^*_1} 1 d\rho_X \\
\int_{\Delta^*_1} x_1 d\rho_X \\
\int_{\Delta^*_1} x_2 d\rho_X \\
0
\end{bmatrix}
= \begin{bmatrix}
\int_{\Delta_B} 1 d\rho_X & \int_{\Delta_L} 1 d\rho_X & \int_{\Delta_R} 1 d\rho_X & \int_{\Delta_*} 1 d\rho_X \\
\int_{\Delta_B} x_1 d\rho_X & \int_{\Delta_L} x_1 d\rho_X & \int_{\Delta_R} x_1 d\rho_X & \int_{\Delta_*} x_1 d\rho_X \\
\int_{\Delta_B} x_2 d\rho_X & \int_{\Delta_L} x_2 d\rho_X & \int_{\Delta_R} x_2 d\rho_X & \int_{\Delta_*} x_2 d\rho_X \\
0 & 1 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
c_0 \\
c_1 \\
c_2 \\
c_3
\end{bmatrix}
\]

and

\[
\begin{bmatrix}
\int_{\Delta^*_2} 1 d\rho_X \\
\int_{\Delta^*_2} x_1 d\rho_X \\
\int_{\Delta^*_2} x_2 d\rho_X \\
0
\end{bmatrix}
= \begin{bmatrix}
\int_{\Delta_B} 1 d\rho_X & \int_{\Delta_L} 1 d\rho_X & \int_{\Delta_R} 1 d\rho_X & \int_{\Delta_*} 1 d\rho_X \\
\int_{\Delta_B} x_1 d\rho_X & \int_{\Delta_L} x_1 d\rho_X & \int_{\Delta_R} x_1 d\rho_X & \int_{\Delta_*} x_1 d\rho_X \\
\int_{\Delta_B} x_2 d\rho_X & \int_{\Delta_L} x_2 d\rho_X & \int_{\Delta_R} x_2 d\rho_X & \int_{\Delta_*} x_2 d\rho_X \\
0 & 1 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
d_0 \\
d_1 \\
d_2 \\
d_3
\end{bmatrix}
\]

We already have an approximation for all of the integrals in these systems. So the systems simplify down to

\[
\begin{bmatrix}
Q_X(\Delta^*_1) \\
0
\end{bmatrix}
= \begin{bmatrix}
Q_X(\Delta_B) & Q_X(\Delta_L) & Q_X(\Delta_R) & Q_X(\Delta_*) \\
0 & 1 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
c_0 \\
c_1 \\
c_2 \\
c_3
\end{bmatrix}
\]

and

\[
\begin{bmatrix}
Q_X(\Delta^*_2) \\
0
\end{bmatrix}
= \begin{bmatrix}
Q_X(\Delta_B) & Q_X(\Delta_L) & Q_X(\Delta_R) & Q_X(\Delta_*) \\
0 & 1 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
d_0 \\
d_1 \\
d_2 \\
d_3
\end{bmatrix}
\]

We solve for the coefficients \(c_0, c_1, c_2, c_3, d_0, d_1, d_2\) and \(d_3\). At this point we run into the same issue as Section 2.1. The systems could be near singular, causing difficulty directly solving. Therefore, we solve the systems just as before, using the truncated SVD for some threshold. Once the coefficients are found, we plug the values into the equations (3.111) and (3.112), giving us \(Q_Y(\Delta^*_1)\) and \(Q_Y(\Delta^*_2)\). From \(Q_Y\) and \(Q_X\) an
approximation of the surface can be constructed. Our focus is on the calculation and preservation of the quantities, not how the quantities are used in the approximation.

**Numerical Results for Distribution-Dependent Subdivision over Triangles**

In this section we present the results of the blending algorithm over the triangles. The theory suggests that the algorithm should perform well for linear functions and regular distributions with gaps of the forms above. For the following tests, the domain $X$ is a $13 \times 13$ square centered at the origin. The initial partition consists of two right isosceles triangles and then is subdivided according to newest vertex bisection. After four levels of uniform subdivision (32 triangles), we insert the points from the point cloud into the triangles and directly calculate the *distribution-dependent* quantities $M_{00}$, $M_{10}$, $M_{01}$, and the *value-dependent* quantity $M_y$. No other part of the algorithm directly accesses the point cloud. Let $T^*$ denote the triangulation where the quantities are calculated directly from the point cloud. The subsequent levels of subdivision calculate the quantities of the children based on the parent’s and parent’s neighbors’ quantities according to the blending algorithm in Section 3.3.

The figures in this section colored red are plots of the average $y$-value ($M_y/M_{00}$) over the triangles. The green figures are the average $x_1$-values ($M_{10}/M_{00}$), and the blue figures are the average $x_2$-values ($M_{01}/M_{00}$).

Our first test is on the point cloud $D_1$, which consists of 653,629 points uniformly distributed over $X$ with a gap between the lines $x_2 = -0.97x_1 + 0.83$ and $x_2 = 0.17x_1 - 1.706$. The $y$-values satisfy the equation

$$y(x_1, x_2) = 2(x_1 - 1)(x_1 + 2) - 3x_2(x_2 + 0.5).$$

Figure 3.30 displays the actual point cloud $D_1$ with the triangulation $T^*$. Figure 3.31 is the average values of the triangles for the triangulation $T^*$ which is the basis for the
Figure 3.30  The point cloud $D_1$ and the triangulation $\mathcal{T}^*$. Note that measure over each can be approximated well by the triangle and complementary representative measures discussed in Section 3.3.

Figure 3.31  The average values over the triangles in the triangulation $\mathcal{T}^*$ and for the point cloud $D_1$.

subsequent subdivisions. The three levels of subdivision of $\mathcal{T}^*$ are shown in Figure 3.32 in which the quantities are calculated from the previous level. As we progress through the different levels of subdivision the algorithm approximates the average $y$ well and the average $x_1$ and $x_2$ extremely well. The gap becomes increasingly apparent with each level of subdivision. Continuing the subdivision to level 10, we get the idea of the limiting surface (Figure 3.33). When comparing the algorithm results Figure 3.33 and the actual point cloud Figure 3.34 we see that the algorithm result is very close to the actual point cloud with the exception of small jagged triangles along the edge of the gap.

The second test is on the point cloud $D_2$, which consists of 376,451 points uniformly distributed over the circle centered at the origin with radius 5. The $y$-values
Figure 3.32 The rows depict the subdivision levels 1 through 3 of $\mathcal{T}^*$. The red plots are the average $y$-value. The green plots are the average $x_1$-values and the blue plots are the average $x_2$-values.

satisfy the linear equation

$$y(x_1, x_2) = 1 + x_1 - x_2.$$  

Figure 3.35 displays the actual point cloud $D_2$ with the triangulation $\mathcal{T}^*$. Figure 3.36 is the average values of the triangles for the triangulation $\mathcal{T}^*$ which is the basis for the subsequent subdivisions. The three levels of subdivision of $\mathcal{T}^*$ are shown in Figure 3.37 in which the quantities are calculated from the previous level. As we progress through the different levels of subdivision the algorithm approximates the average $y$ well and the average $x_1$ and $x_2$ extremely well. Continuing the subdivision
Figure 3.33 Subdivision level 10 (the limit surface).

Figure 3.34 The point cloud $D_1$ colored red according to the average $y$-value, colored green according to the average $x_1$-value, and then colored blue according to the average $x_2$-value.

Figure 3.35 The point cloud $D_2$ and the triangulation $\mathcal{T}^*$. The better these initial measures over the triangles are represented by the triangle and complementary measures discussed in Section 3.3, the better the limiting surface of the algorithm.
Figure 3.36 The average values over the triangles in the triangulation $T^*$ and for the point cloud $D_2$.

Figure 3.37 The rows depict the subdivision levels 1 through 3 of $T^*$. The red plots are the average $y$-value. The green plots are the average $x_1$-values and the blue plots are the average $x_2$-values.
to level 10, we get the idea of the limiting surface (Figure 3.38). When comparing the algorithm results Figure 3.38 and the actual point cloud Figure 3.39 we see that the algorithm result is very close to the actual point cloud with the exception of small jagged triangles along the edge of the circle. Most of the error in the average \( y \)–values occurs around the edges of the circle.

**Remarks**

In this section we consider a case that requires a more diverse selection of representative measures to approximate the measure well. Let the measure \( \rho \), of the *reference triangle* \( \Delta^* \), be equivalent to the Lebesgue measure over the triangle

\[
A = \left( (-5/12, 0), (-10/51, 27/68), (1/3, 0) \right).
\]
Figure 3.40. The image on the left depicts the triangle with Lebesgue measure over $A$. The image on the right depicts the subdomain representative measure found by the algorithm. This poor representation leads to visible errors as we refine.

Note that $A$ is not contained in $D$ and $d\rho = \chi_A \, dx$. Then the \textit{distribution-dependent} quantities are approximately

$$M_{00}(\Delta^*, \rho) = .14890, \quad M_{10}(\Delta^*, \rho) = -0.01386, \quad \text{and} \quad M_{01}(\Delta^*, \rho) = .0049786.$$

Thus $q(\rho) = (-0.093137, 0.033437)$ lies in the region $T_3$ on the $uv$-plane, which corresponds to a Type 3 complementary measure. Following the calculations described in Section 3.3 for the computation of the complementary measure parameters, we have that $b = .95463$ and $c = -.91433$. Figure 3.40 displays both the measure $\rho$ and the representative measure. The measure $\rho$ is not approximated well by the subdomain representative measure. Furthermore, this (or similar) issue arises with any characteristic measure significantly different from our triangle measures and complementary measures. We are merely laying the foundation for work in this area and expected this type of error. Future work will explore higher order schemes which contain more diverse representative measures.

The initial level of subdivision greatly impacts the limiting surface of the subdivision. If the measure of the initial level can be represented well by the triangle complementary measures in Section 3.3, then the limiting surface of the algorithm will also be good. On the other hand if the measure is poorly approximated initially, then that error will propagate through the refinement. Consider the uniform point
Figure 3.41  The first plot is the average $y$-values of the rectangular point cloud with $T^*$. The plot in the middle is the algorithm result after 5 additional levels of subdivision. The third plot is after 15 levels of subdivision. Notice the error around the corners of the rectangle.

cloud on the rectangle

$$R = ((-3, -4), (-3, 4), (3, 4), (3, -4))$$

with $y = 1 + x_1 - x_2$. Then 4 triangles in $T^*$ encounter the same corner issue discussed above, and that error is significant 5 levels of subdivision later and continues to the limit surface (Figure 3.41). Future work will explore higher order schemes which contain more diverse representative measures.
CHAPTER 4

CONCLUSION

In this dissertation we address the issues of irregularly distributed point clouds from two different perspectives. In Chapter 2 we address several critical issues arising in practical implementations of processing real point cloud data that exhibits irregularities. We examine some ideas known to work well for regular point clouds and modify them to extend their applicability for realistic raw data with a semi-regular underlying topology. We develop practical algorithms that realize certain ideas based on the theory and methodology in Learning Theory using adaptive partitioning from [4] and [2]. In particular, we address the issues of instability for higher order polynomial approximation described in [2] and suggest practical algorithms using truncated SVD to ensure that our approximation does not encounter this problem and to add to the stability. We employ our algorithm in the analysis and processing of LiDAR point clouds. In this realization, special criteria are developed for arriving at a nearly optimal adaptive partition for representation of the surface of interest; different adjustments to the results are applied to enhance the results and increase stability, including clustering to separate different sensed objects in the calculation of the approximation. A new nonlinear encoding process that progressively describes the surface approximations is designed for giving a realistic representation, especially for very large compression rates.

Chapter 3 develops the theory and special algorithms targeted at representing curves and surfaces with gaps. These are particularly important for the representation of the approximation results of the types of point cloud considered in Chapter 2.
While the standard subdivision schemes are developed to apply for regular distributions, we suggest a new approach that gives the opportunity to extend this important visualization tool to new practically important areas. The algorithms analyze the aggregate quantities from the coarser level, separating them into distribution-dependent and value-dependent, to calculate their counterparts at a finer level. We give a groundwork for further theoretical developments in this area and the exploration of new distribution-dependent subdivision schemes.

4.1 Future Work

Here we describe our considerations for future work in the area.

Related to the research discussed in Chapter 2 we have the following topics: modification of the spherical learning algorithm to process point clouds from moving sensors; extension of the algorithm beyond \( d = 2 \) by partitioning with general simplices; and development of more sophisticated compression techniques to improve our rates.

The ideas in Chapter 3 can be extended so that we have reproduction of higher order moments. In doing so, we will be able to have perfect recovery of more diverse types of measures. Along the same lines, a more sophisticated blending scheme could be implemented. We have already started exploring the polynomial of degree three weight measures and the corresponding subdomain measure.
Bibliography


