EFFECTS OF GRID LATTICE GEOMETRY ON DIGITAL IMAGE FILTERING

by

Roger Owen Brown

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Civil Engineering

APPROVED:

Dr. Steven Johnson
Co-Chairperson

Dr. Clifford Hottman
Co-Chairperson

Dr. Robert McEwen

Mr. David Scopp

April, 1989

Blacksburg, Virginia
EFFECTS OF GRID LATTICE GEOMETRY ON DIGITAL IMAGE FILTERING

by

Roger Owen Brown

Committee Chairpersons:  Dr. Steven Johnson
                       Dr. Clifford Kottman

(ABSTRACT)

The spatial distribution of discrete sample points from an image affect digital image manipulation.

The geometries of the grid lattice and edge are described for digital images. Edge detecting digital filters are considered for segmenting an image. A comparison is developed between digital filters for two different digital image grid lattice geometries -- the 8-neighbor grid lattice (rectangular tesselation) and the 6-neighbor grid lattice (hexagonal tesselation). Digital filters for discrete images are developed that are best approximations to the Laplacian operator applied to continuous two-dimensional mathematical surfaces. Discrepancies between the calculated Laplacian and the digital filtering results are analyzed and a criterion is developed that compares grid lattice effects. The criterion shows that digital filtering in a 6-neighbor grid lattice is preferable to digital filtering in an 8-neighbor grid lattice.
I would like to express appreciation to the following people for their support. Dr. Clifford Kottman supported this thesis with continual reviews and suggestions while the project developed. His patient reviews and creative suggestions made him an invaluable technical advisor. Dr. Steven Johnson reviewed and edited the manuscript as an academic advisor. Dr. Robert McEwen and Mr. David Scopp reviewed the manuscript as technical advisors. The Defense Mapping Agency provided the financial support for me to attend graduate school. Virginia Polytechnic Institute’s Northern Virginia Graduate Center provided the Civil Engineering graduate curriculum. I owe special appreciation to my wife Robina and our children for allowing me to concentrate on studies while attending the program.
TABLE OF CONTENTS

Abstract ........................................... ii
Acknowledgements ................................... iii
List Of Figures ..................................... viii
List Of Tables ...................................... ix

1. Introduction ...................................... 1
   1.1 Problem Statement ............................. 1
   1.2 Objectives .................................... 1

2. Digital Image Processing Background ........ 3
   2.1 Definitions ................................... 3
   2.2 Edge Detection Applications ................. 8
   2.3 Digital Sampling ................................ 9
      2.3.1 Image Digitizing Procedure ................ 9
      2.3.2 Grid Lattice Geometry ..................... 10
      2.3.3 Sampling Density ........................... 16
      2.3.4 Image Rectification ....................... 18
   2.4 Literature Review ............................. 19

3. Development Of Digital Laplacian Filters .... 22
   3.1 Digital Filter Matrix Convolution .......... 23
      3.1.1 Discrete Difference Methods .............. 23
      3.1.2 Matrix Convolution Formulation ............ 27
      3.1.3 Grid Post Spacing Weight Matrix ........... 31
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>File Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.3</td>
<td></td>
<td>Surface Of Rotation Differentials</td>
<td>85</td>
</tr>
<tr>
<td>A.4</td>
<td></td>
<td>Arctangent Edge Profile 2nd Derivative Roots</td>
<td>86</td>
</tr>
<tr>
<td>A.5</td>
<td></td>
<td>Grid Lattice Density</td>
<td>87</td>
</tr>
<tr>
<td>B.1</td>
<td></td>
<td>Computer Files</td>
<td>90</td>
</tr>
<tr>
<td>B.2</td>
<td></td>
<td>Introduction</td>
<td>90</td>
</tr>
<tr>
<td>B.2.1</td>
<td></td>
<td>EDGPRO.EXE</td>
<td>92</td>
</tr>
<tr>
<td>B.2.2</td>
<td></td>
<td>EDGPRO.FOR</td>
<td>92</td>
</tr>
<tr>
<td>B.2.3</td>
<td></td>
<td>SURFER.EXE</td>
<td>94</td>
</tr>
<tr>
<td>B.2.4</td>
<td></td>
<td>GRIDDS.INC</td>
<td>94</td>
</tr>
<tr>
<td>B.2.5</td>
<td></td>
<td>SURFER.FOR</td>
<td>95</td>
</tr>
<tr>
<td>B.2.6</td>
<td></td>
<td>CNVLTN.FOR</td>
<td>98</td>
</tr>
<tr>
<td>B.2.7</td>
<td></td>
<td>DENSIT.FOR</td>
<td>102</td>
</tr>
<tr>
<td>B.2.8</td>
<td></td>
<td>SURF.FOR</td>
<td>103</td>
</tr>
<tr>
<td>B.2.9</td>
<td></td>
<td>FILTER.FOR</td>
<td>104</td>
</tr>
<tr>
<td>B.2.10</td>
<td></td>
<td>ELWIND.EXE</td>
<td>105</td>
</tr>
<tr>
<td>B.2.11</td>
<td></td>
<td>ELWIND.FOR</td>
<td>105</td>
</tr>
<tr>
<td>B.2.12</td>
<td></td>
<td>RADIAL.EXE</td>
<td>106</td>
</tr>
<tr>
<td>B.2.13</td>
<td></td>
<td>RADIAL.FOR</td>
<td>107</td>
</tr>
<tr>
<td>B.3</td>
<td></td>
<td>Data Files</td>
<td>108</td>
</tr>
<tr>
<td>B.3.1</td>
<td></td>
<td>EDGPRO.SAS</td>
<td>108</td>
</tr>
<tr>
<td>B.3.2</td>
<td></td>
<td>PATH.SAS</td>
<td>113</td>
</tr>
<tr>
<td>B.3.3</td>
<td></td>
<td>ELWIND.SAS</td>
<td>119</td>
</tr>
<tr>
<td>B.3.4</td>
<td></td>
<td>RADIAL.SAS</td>
<td>119</td>
</tr>
<tr>
<td>B.3.5</td>
<td></td>
<td>TEST.DAT</td>
<td>124</td>
</tr>
</tbody>
</table>
C. Digital Filtering . . . . . . . 128
  C.1 Nth Order Differences. . . . . . 128
  C.2 Conversion Between Cartesian Position And Array Address . . . . . . . 131
  C.3 Sample Digital Filter Matrix Convolutions. . 134
  C.4 Radial Profiles Perpendicular To Edge Lines . 136
  C.5 Frequency Domain Digital Filters . . . 138
  C.6 Grid Lattice Scanner Designs . . . 140

Vita . . . . . . . . . . . . . 144
LIST OF FIGURES

2.1 Grid Lattice Geometry ........................................ 12
3.1 Digital Step Edge ............................................. 34
3.2 Filtered Digital Step Edge .................................... 36
3.3 Continuous Ramp Arctangent Derivatives ..................... 39
3.4 Arctangent Derivatives Versus Differences .................. 41
3.5 Radial Arctangent Profile's Surface Of Rotation ............ 43
4.1 Circular Edge Line Sample Post Orientation ................. 59
4.2 6-Neighbor And 8-Neighbor Grid's Elementary Window ....... 63
4.3 Digital Laplacian Filter Discrepancy Curves ................. 66
C.1 6-Neighbor Grid Lattice N=2 Window ........................ 130
C.2 Radial Profile Digital Laplacian Differences ............... 137
C.3 Scanner Spot Designs ......................................... 142
C.4 6-Neighbor Grid Lattice Scanning Patterns ................. 143
LIST OF TABLES

2.1 Preferential Direction Displacement Vectors . . 13
2.2 Displacement Vector Index Array Matrices . . 14
3.1 Digital Laplacian Filter Matrices . . . 51
4.1 Digital Laplacian Discrepancy Curve Characteristics 70
A.1 Equal Density Post Spacing Distances . . . 89
1. INTRODUCTION

1.1 Problem Statement

A digital image may be considered as a collection of discrete samples of an analogue image. Each sample quantizes image brightness at a sample point. The sample points are usually in a regular geometrical arrangement that is called a grid lattice. Different grid lattices can be designed for a given image resulting in different sample point spatial distributions. Two popular arrangements are the rectangular tesselation (called an 8-neighbor grid lattice) and the hexagonal tesselation (called a 6-neighbor grid lattice) of the image plane.

Digital images enable the analytical manipulation of the original image. Important examples are the edge enhancing or edge finding operators, where edges are lineal boundaries that divide bright and dark regions of the image.

1.2 Objectives

This objective of this thesis is to compare the performance of the most common edge operator, the Laplacian convolution, on digital images of the 8-neighbor grid lattice versus the 6-neighbor grid lattice. We introduce new forms of the discrete Laplacian convolution to accomplish filtering of the digital image. Then we execute a test that does a comparison between the digital Laplacian
filters for the two grid lattice point geometries.

This thesis has several procedural objectives which describe the sample point spatial distribution’s digital filtering effects. The relevance of digital filtering to digital image processing is described. Different image plane sample point spatial distributions are compared. The digital filtering concept is developed to accomplish matrix convolution on the digital image. The Laplacian is justified to discover image edges. Equivalent Laplacian convolutions are developed for different digital image grid lattices. Array structures and mathematical operations are developed to accomplish digital filter matrix convolution. A test is designed that will compare the equivalent digital Laplacian convolutions in different grid lattices. The test results will quantify the grid lattice preference while using matrix convolution to do edge operations on the digital image.
2. DIGITAL IMAGE PROCESSING BACKGROUND

In this chapter, the relevance of digital filtering to image processing is described. Finally, the digital image's grid lattice geometry is described.

2.1 Definitions

An image can be considered as a continuous real function of an image plane's spatial \((x,y)\) position. Let \(f(x,y)\) be a brightness value on the image plane. The \([x,y,f(x,y)]\) coordinates form a continuous two-dimensional mathematical surface, so that \(f(x,y)\) may be considered as the image brightness surface. The image brightness surface's continuity is affected by the sensor's transfer of the scene in object space to the image plane in imaging space. For all practical purposes, the image can be considered continuous.

Tesselation partitions the plane into edge matched polygons. For example, models of atomic arrangements known as quasi-crystals describe the partitioning of a plane by edge matched parallelograms. A simple tesselation method will completely tile the plane with only one type of polygon and with no gaps between any one of the polygon tiles. The tesselated plane's polygon shape is described by the length of the polygon sides and by angles at the polygon side's vertexes. This paper will only consider tesselations
formed by an equiangular quadrilateral (square or rectangle) or an equiangular hexagon.

The rectangular and hexagonal grid lattices were utilized in this paper's experimental design because they were the most economical ways to tesselate the image plane. Rectangles and triangles minimize the number of tesselating polygon's sides. Less polygon sides produce a simple geometric description for the spatial area that surrounds each sample point. The hexagonal tesselation is the one repeating pattern that uses the least number of lines to cover a given area. Therefore, hexagons are physical structures that naturally occur as honeycomb patterns. For example, hexagonal tesselation occurs when soap film bubbles cover a plane while each individual bubble is trying to be as small as possible. The grid lattice formed by the triangular versus hexagonal tesselation are similar because a conjugate group of six equilateral triangles forms a regular hexagon.

A grid lattice is a spatial distribution of sample points on a plane. Each grid post is identified with a sample value that is located at a spatial \((x,y)\) position on the plane. Each grid post is a vector that may be perpendicular to the plane. The vector's magnitude may be determined by the \(f(x,y)\) sample value at the sample point's spatial position, \((x,y)\). Each vector's initial point is a sample point that is surrounded by a polygon. The
vector's initial point usually is the polygon's centroid. The vector endpoints describe the mathematical surface of sample values.

The image plane is tessellated with identical conjugate polygon shaped partitions that are called pixels (picture elements). Pixels usually are square or rectangular shaped polygons. Each pixel contains one grid post that is the pixel's centroid. Each grid post may have a quantized image attribute such as a brightness value assigned to it from the surrounding pixel.

The grid lattice geometry describes the sampling point spatial distribution on the plane. The spatial distribution can be described with displacement vectors between adjacent grid posts. The grid post spatial distribution is an incidental byproduct of the tessellation of the plane by polygons. An elementary window is a small group of grid posts that sufficiently explain the whole grid lattice's geometry. A symmetric grid lattice has equally spaced grid posts in fixed displacement vector directions. An edge's geometry is described by a line's shape, orientation, and sharpness in the image plane's spatial domain.

The image transfer function transfers the imaged object's brightness to the image plane. Brightness is quantized from the image transfer function's energy amplitude. Quantization subdivides a continuous quantity of energy into measurable discrete increments. Brightness
may be quantized with discrete gray levels.

At each grid post position, \((x,y)\), the image brightness function, \(f(x,y)\), determines an aggregation of the brightness level for each corresponding pixel. Other times, a grid post is mapped to the evaluation of a continuous function at just the post's location, rather than an average over the pixel. The latter procedure is used in this thesis to digitize the mathematical surface.

The digital image is a discrete form of the image that can be stored in matrix formats and can be numerically processed. A digital filter is a matrix composed of discrete values of a point spread function that spans a collection of adjacent grid posts. Digital filter matrix convolution occurs when the digital filters are convolved on the digital image through a special mathematical operation that produces a filtered digital image. Digital convolution is an analog to continuous function convolution. The digital filters are commonly called the kernels of the convolution. Section 3.1.2 describes digital filter matrix convolution on the digital image's matrix.

We will use ordinary mathematical tools to study the image brightness function, \(f(x,y)\). Image brightness moments are derivatives of the image brightness function. Brightness moments can be expressed in terms of the brightness function's \(i^{\text{th}}\)-order derivatives, \(\partial^i f(x,y)/\partial x^i\) or \(\partial^i f(x,y)/\partial y^i\). In particular, we use the first-order and
second-order derivatives to study image brightness moments. Gradients use the first-order derivatives (slopes) of the image plane's brightness function,
\[ \text{grad}(f) = \nabla f = \left( \frac{\partial f}{\partial x} \right) \mathbf{i} + \left( \frac{\partial f}{\partial y} \right) \mathbf{j} = [\delta f/\delta x, \delta f/\delta y]. \]

The brightness gradient is an elementary measurement of brightness spatial distributions on the image plane. The brightness gradient divides light and dark brightness regions on the image plane. We will be interested in image regions where the image brightness function's first-order derivatives are nearly constant. Edges are abrupt changes in the brightness gradient. Relatively large second-order derivatives of the image brightness function are used to find the edges where abrupt brightness gradient changes are occurring.

Slopes on the continuous image's brightness surface are replaced by grid post brightness value difference steps between adjacent grid posts on the image plane. Grid post value difference steps are measured by subtracting quantized brightness vector magnitudes that exist between adjacent grid posts. \( i^{th} \)-order differences, \( f^{(i)} \), on the digital image will replace \( i^{th} \) order derivatives, \( f^{(i)} \), on the continuous image.

Regions will exist within the digital image where the grid post brightness value difference steps, between adjacent grid posts, are small or predictable. The image can be segmented by determining boundaries between those
regions. The boundaries may be indicated by large thresholded second-order difference values. An edge is determined in the digital image when second-order difference thresholds are exceeded.

2.2 Edge Detection Applications

Edges can be detected by matrix convolution operations which apply digital filters to the digital image's matrix of brightness values.

Edge detection may be used in imagery analysis to segment the digital image into brightness regions of different tonal roughness. Tonal roughness is determined by brightness spatial frequencies within the image. Brightness spatial frequencies may be described by gray level distributions within the digitized image.

Edges delineate regions with simple curves (polygons or arcs). Those edges show where abrupt changes occur in the gray level distribution within each delineated region on the image plane.

Statistical methods can attach feature identification probabilities to various gray level distributions within delineated regions on the image plane, assuming each feature group has homogenous reflectance or emittance properties within the image. Rosenfeld & Kak (1982) present various brightness distribution properties including gray level variance, co-occurrence matrices, coarseness histograms, plus
autocorrelation and power spectrum. Lillesand & Kiefer (1979) present basic spectral pattern recognition strategies to identify and classify brightness regions within the image.

The gray level distribution within each brightness region can be associated with other brightness regions within the image. Brightness regions with similar gray level distribution may have the same image characteristics for feature identification purposes.

Edge detection techniques can be applied to digital elevation models. Geomorphic features such as ridges and potential drainage troughs can be discovered by applying digital filters to the terrain surface that is formed by f(x,y) elevation values in a rectangular coordinate system.

2.3 Digital Sampling

A digital image is a collection of discrete brightness value samples on the image plane. This section describes methods for, and results from, digitally sampling an image.

2.3.1 Image Digitizing Procedure

Various sensors exist to convert energy levels into pixels with discrete gray levels.

An image digitizer is composed of a scanner that uses sensors to measure and quantize energy levels (usually brightness values) that transfer from the scene to the
image. The actual scene may be directly digitized by the scanner, or the image may be digitized after the scene is exposed on film. This paper assumes a simple response to brightness by sensors inside the scanner. Castleman (1979) describes image digitizing system designs.

Image digitizing is a discrete process, where each sensor spot maps a quantized brightness value to a single pixel gray level on the image plane. The number of discrete brightness values (or gray levels) per pixel depends on the amount of computer storage assigned to each pixel. For example, 8-bit pixel data can quantize up to 256 ($2^8$) discrete brightness values.

2.3.2 Grid Lattice Geometry

A grid lattice's sampling post spatial distribution may be described by displacement vectors between adjacent grid posts on the image plane. The displacement vectors describe the geometric pattern of the grid posts.

An elementary window of the grid lattice is defined in terms of one central grid post and a certain number of that post's nearest neighboring posts. An $n$-neighbor grid lattice is an elementary window of $n+1$ posts. An elementary window is large enough to explain the whole grid lattice's geometry.

Processing of the grid lattice's digital data requires that the grid post patterns are described by an array
storage function. Index arrays may contain displacement vectors that describe the positional relationship between the grid lattice window's central post and each of its neighboring posts within the elementary window. The grid post spacing distance is the displacement vector's magnitude. Preferential directions are the displacement vector's directions.

Two common grids are the 8-neighbor grid lattice and the 6-neighbor grid lattice. Each grid lattice has different displacement vector index arrays. Figure 2.1 compares the two grid lattices within each grid lattice's elementary window. Each grid post is enclosed by a rectangle in both grid lattices.

Table 2.1 shows the displacement vectors for each grid lattice. The 8-neighbor grid lattice's elementary window is a rectangular array of nine posts that yield four displacement vectors we shall label as \( \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{\ell} \). The 6-neighbor grid lattice's elementary window is a hexagonal array of seven posts that yield three displacement vectors we shall label as \( \mathbf{u}, \mathbf{v}, \mathbf{w} \). The 4-neighbor grid lattice is a special case of the 8-neighbor grid lattice without the \( \mathbf{g} \) and \( \mathbf{t} \) displacement vectors, no diagonal posts, in the elementary window. Table 2.2 shows the 3x3 displacement vector index arrays that are necessary to contain each of the three grid lattice's displacement vectors.
6-Neighbor Grid Directions

8-Neighbor Grid Directions

Figure 2.1: Grid Lattice Geometry
Table 2.1: Preferential Direction Displacement Vectors

8-Neighbor Grid Lattice

\[ \mathbf{x} = [ \Delta X, 0.0 ] \]
\[ \mathbf{y} = [ 0.0, \Delta Y ] \]
\[ \mathbf{z} = [ \Delta X, -\Delta Y ] \]
\[ \mathbf{t} = [ \Delta X, \Delta Y ] \]

6-Neighbor Grid Lattice

\[ \mathbf{u} = [ (\Delta X) \sin(30^\circ), -(\Delta X) \cos(30^\circ) ] \]
\[ \mathbf{v} = [ \Delta X, 0.0 ] \]
\[ \mathbf{w} = [ (\Delta X) \sin(30^\circ), (\Delta X) \cos(30^\circ) ] \]

4-Neighbor Grid Lattice

\[ \mathbf{x} = [ \Delta X, 0.0 ] \]
\[ \mathbf{y} = [ 0.0, \Delta Y ] \]

Note: \( \Delta X \) and \( \Delta Y \) are displacements in the coordinate system's orthogonal directions, \( \mathbf{x} \) and \( \mathbf{y} \) in Figure 2.1.
Table 2.2: Displacement Vector Index Array Matrices

**8-Neighbor Index Array**

\[
\begin{bmatrix}
(-\Delta X, -\Delta Y) & (-\Delta X, 0.0) & (-\Delta X, \Delta Y) \\
(0.0, -\Delta Y) & (0.0, 0.0) & (0.0, \Delta Y) \\
(\Delta X, -\Delta Y) & (-\Delta X, 0.0) & (\Delta X, \Delta Y)
\end{bmatrix}
\]

**6-Neighbor Index Array**

\[
\Delta'X = (\Delta X)\sin(30^\circ) = (\Delta X)/2 \\
\Delta'Y = (\Delta X)\cos(30^\circ)
\]

\[
\begin{bmatrix}
(-\Delta'X, -\Delta'Y) & (-\Delta X, 0.0) & (-\Delta'X, \Delta'Y) \\
empty & (0.0, 0.0) & empty \\
(\Delta'X, -\Delta'Y) & (\Delta X, 0.0) & (\Delta'X, \Delta'Y)
\end{bmatrix}
\]

**4-Neighbor Index Array**

\[
\begin{bmatrix}
empty & (-\Delta X, 0.0) & empty \\
(0.0, -\Delta Y) & (0.0, 0.0) & (0.0, \Delta Y) \\
empty & (\Delta X, 0.0) & empty
\end{bmatrix}
\]
Angular resolution is described by angles between the displacement vectors in each grid lattice's elementary window of grid posts. Figure 2.1 shows angles between the displacement vectors in each elementary window. The angular resolution is 90° for the 4-neighbor grid lattice, 60° for the 6-neighbor grid lattice, and 45° for the 8-neighbor grid lattice.

A symmetric grid lattice has equally spaced posts and angular symmetry. Equally spaced grid posts occur when all of the elementary window's index array displacement vector magnitudes are equal. Angular symmetry occurs when there is constant angular resolution between displacement vectors within the grid lattice's elementary window. The 4-neighbor grid lattice and 6-neighbor grid lattice can be symmetric if all their displacement vector magnitudes are equal. The 8-neighbor grid lattice has greater angular resolution but all its displacement vector magnitudes are not equal. Therefore, the 8-neighbor grid lattice only has angular symmetry. It follows that angular symmetry is necessary but not sufficient to establish a symmetric grid lattice.

An 8-neighbor grid lattice can be transformed to a 6-neighbor grid lattice by shifting alternating columns of the 8-neighbor grid posts by one-half the grid post spacing distance along alternating columns in the ΔX direction (during image digitizing), to maintain the density of the transformed grid lattice. But a symmetric 6-neighbor grid
lattice will not result unless the ratio of the \( u \) or \( w \) displacement vector elements, \((\Delta'X, \Delta'Y)\), in the displacement vector index array is

\[
\frac{|\Delta'Y|}{|\Delta'X|} = \tan 60^\circ = (3)^{1/2}
\]

(2.1)

Pixel shape and dimensions affect image resolution. The symmetric 6-neighbor grid lattice can completely cover the image plane with rectangular pixels if the condition in Equation 2.1 holds. Therefore, the resolutions of the 6-neighbor grid lattice will be different in the two orthogonal directions of image space (i.e. \( \Delta X \) does not equal \( \Delta Y \)), unless the pixels are equiangular hexagons. This thesis ignores pixel shape and dimension effects on resolution by just evaluating the image brightness function, \( f(x,y) \), at each grid post's \((x,y)\) position.

2.3.3 Sampling Density

The grid lattice sampling post density is a geometric consideration. The grid lattice density is the expected number of sampling posts in a window of unit area on the image plane. The grid lattice density can be determined by assuming one grid post per rectangular pixel, where there are no gaps between any of the pixels covering the image plane. Pixel areas can be equated between different grid lattices to equate grid lattice densities of sampling posts.
A digital image should more faithfully represent the sampled continuous image as the grid lattice density increases, regardless of the sample point spatial distribution on the image plane.

Each grid lattice implies a sampling of the continuous image at discrete image plane positions. The placement of the grid lattice with respect to a particular edge's shape, orientation, and sharpness in the image affects the digital image's ability to detect that edge.

Ideally, an edge should possess some measurable qualities throughout the image, such as a thresholded range of quantized values when an appropriate mathematical operator is applied to the image brightness surface. However, an edge may hit or miss sample posts as the edge threads through the grid lattice. Geometric relationships between the edge's geometry (shape, orientation, and sharpness) and the grid lattice's geometry (grid post spatial distributions) will affect the probabilities of the edge line hitting, or missing, grid posts.

The grid lattice density also affects the digital image's ability to detect an edge. The probabilities of the edge's path hitting posts should increase as grid lattice density increases, regardless of grid lattice geometry.

The grid lattice densities should be equal between all compared grid lattices, so the effects of the grid lattice sampling post spatial distribution can be isolated.
2.3.4 Image Rectification

Image rectification removes various imagery distortions. A digital image can be rectified without being encumbered by the mechanical limitations of optical or correlation hardware. Digital imagery minimizes the restrictions on image rectification because there are practically no limits on analytical methods to manipulate the data. Thus more imagery distortions can be removed from digital imagery. Less imagery distortions will improve the imagery analysis quality, particularly if a stereomodel is being utilized.

It may be desirable to remove noise before filtering the digital image, an aspect of image restoration. Noise will occur while the sensor is processing the signal emanating from the scene. The Manual of Remote Sensing (1983) describes sources of signal dependent noise and signal independent noise. Conventional texts (Castleman 1979, Rosenfeld and Kak 1982) classify numerous types of noise, and even more numerous methods for removing the noise from the image.

It may be desirable to remove noise from a digital image before digital filters are applied in order to eliminate the effect of noise on edge detection. In fact, digital filter matrix convolution may be used to remove noise from the actual digital image. The functions to remove noise from the image may have to be picked by trial
and error unless you have some prior knowledge of the noise existing in the image.

The issue of image rectification, particularly noise removal, is outside the scope of this thesis. Image rectification issues are avoided through the choice of an ideal continuous mathematical surface that contains recognized edges to compare grid lattice edge detection capabilities.

2.4 Literature Review

Increases in preferential directions with no directional or connectivity ambiguity (e.g., in the 4-neighbor grid lattice versus the 8-neighbor grid lattice with its additional diagonal directions) and angular resolution are frequently cited as justification for a 6-neighbor grid lattice versus an 8-neighbor grid lattice. Crettz (1980) developed a cosine transformation for 6-neighbor grid lattice digital filtering in the frequency domain. However, he did not develop digital filters for image filtering in the spatial domain of the image.

Van-Roessal (1988) presents conversion algorithms between hexagonal grid arrays and the Cartesian position. His spatial addressing method arranges hexagonal pixel bunches into hierarchical aggregates, where there is constant angular resolution between connected aggregations of pixels for each aggregation level. Such Generalized
Balanced Ternary (GBT) addressing schemes may be useful for image minification (where pixel values must be bunched and assigned average brightness values plus a position in the new image). He admits those addressing methods are inefficient; this thesis presents an alternative addressing method in the Appendix that does not accommodate image minification.

Certain derivative operations are preferable when considering edge orientations with respect to the grid lattice geometry. Rotation invariance (often called isotropy) is chosen as the desirable characteristic of derivative operations because the edge's orientation is unpredictable. Digital filters must use differencing operations to approximate derivative operations. So rotation invariant derivative operations should be mimicked by digital filters. Rosenfeld & Kak (1982) present those issues but they do not quantify the digital filtering effects.

There are unlimited methods to partition the plane into edge matched polygons, called tesselation. The problem of partitioning or tiling a plane is an old mathematical problem that started received renewed attention in the mid-1970s. Rucker (1987) describes the geometric qualities of tesselating a plane. Ivars Peterson (1988) summarizes the work that is currently being done to develop parallelogram edge matching rules for tesselating a plane.
McEachren (1982) and Fairchild (1981) attempted to determine grid structure preferability by changing the pixel shape surrounding each grid post. Different grid post spatial distributions were an incidental result of their various pixel shapes. Results that compared map accuracy for a variety of grid orientations with respect to a set of different elevation surface complexities (i.e., terrain roughness) were inconclusive.
3. DEVELOPMENT OF DIGITAL LAPLACIAN FILTERS

The following items are included in this chapter. The n-neighbor grid lattice’s digital Laplacian filter is developed. The concept of digital filter matrix convolution is developed to compute discrete differences that approximate derivatives on continuous surfaces. Inverse distance weight matrices are introduced to account for the effects of grid post spacing. Simple edges are described and defined in terms of their derivatives and differences. An arctangent function with horizontal and vertical scalars (controlling the edge sharpness) is presented as a continuous representation of the discrete step edge. A mathematical surface of rotation is developed from the arctangent function’s radial profile. Then the Laplacian’s edge detection utility is justified because of rotation invariance. The mathematical surface of rotation is presented as a test for digital filter matrix convolution because of recognized circular edge lines with constant mathematical Laplacian values. Equivalent digital Laplacian filters are developed for various grid lattices from matrix qualities noticed in the 4-neighbor grid lattice’s digital Laplacian filter.

All those formulations satisfy an approach to designing and comparing digital filter matrix convolution for different grid lattices. The digital filters will remove
all possible grid lattice error sources, except preferential directions that are inherent in each grid lattice’s digital filter. All this chapter’s formulations will be used to design an experiment that compares grid lattice geometry effects during edge detection, using the digital Laplacian filter on ideal continuous two-dimensional mathematical surfaces of rotation.

3.1 Digital Filter Matrix Convolution

Convolution is a mathematical sampling of a two-dimensional mathematical surface, \( f(x,y) \). That surface may be formed by brightness values on the image plane. Digital filter matrix convolution uses differencing operations to estimate derivatives. The formulations in Section 3.1 provide the mathematical basis for the development of the digital Laplacian filter.

3.1.1 Discrete Difference Methods

Edges occur when there are abrupt changes in the brightness gradient along the image brightness surface. Large second-order derivative values will indicate abrupt changes in a continuous image’s gradient. The magnitude of those abrupt gradient changes determines the edge sharpness. A digital image will indicate abrupt changes on the image brightness surface by unexpected grid post value difference steps between adjacent grid posts. The edge will be
indicated by large second-order difference steps. Such anomalies will be considered edge boundaries for regions of predictable brightness gradient, or grid post value difference stepping patterns, within the image.

Digital filtering approximates derivatives by using differences between adjacent grid posts that surround the derivative's grid post. Letting the grid post's spatial position be indicated by \((x,y)\), or \((u)\), the derivative's definition is

\[
\frac{\delta f(u)}{\delta u} \delta(U) = \lim_{\Delta U \to 0} \frac{f(U+\Delta U) - f(U)}{\Delta U}
\]

\[
= \lim_{\Delta U \to 0} \frac{f(U) - f(U-\Delta U)}{\Delta U}
\]

\[
\frac{\partial f(x,y)}{\partial x} \delta(X,Y) = \lim_{\Delta X \to 0} \frac{f(X+\Delta X,Y) - f(X,Y)}{\Delta X}
\]

\[
= \lim_{\Delta X \to 0} \frac{f(X,Y) - f(X-\Delta X,Y)}{\Delta X}
\]

\[
\frac{\partial f(x,y)}{\partial y} \delta(X,Y) = \lim_{\Delta Y \to 0} \frac{f(X,Y+\Delta Y) - f(X,Y)}{\Delta Y}
\]

\[
= \lim_{\Delta Y \to 0} \frac{f(X,Y) - f(X,Y-\Delta Y)}{\Delta Y}
\]
Equations 3.1 show the relationship between differences and derivatives. Digital filters use grid post value difference steps between adjacent grid posts to approximate derivatives. The grid post spacings are specified by $\Delta X$, $\Delta Y$, and $\Delta U$ spatial distances in the $x$, $y$, and $u$ directions. The grid post spacings of $\Delta X$, $\Delta Y$, or $\Delta U$ will never approach the limiting value of $X=0$, $Y=0$, or $U=0$. So the differences will only approximate the derivatives.

An unbiased differencing operation occurs when the differencing operations are symmetric around the convolution window's central post, $f(X,Y)$. Equations 3.2 through 3.3 are the unbiased first-order and second-order differences at position $(x,y)$ for the $\Delta X$ and $\Delta Y$ orthogonal directions, and at position $(u)$ for the $\Delta U$ direction.

\[
\begin{align*}
 f^{(1)}(U) &= \frac{f(U+\Delta U) - f(U-\Delta U)}{2\Delta U} \quad (3.2a) \\
 f_x^{(1)}(x,y) &= \frac{f(x+\Delta X,y) - f(x-\Delta X,y)}{2\Delta X} \quad (3.2b) \\
 f_y^{(1)}(x,y) &= \frac{f(x,y+\Delta Y) - f(x,y-\Delta Y)}{2\Delta Y} \quad (3.2c)
\end{align*}
\]
\[
\frac{f(U+\Delta U) - f(U)}{\Delta U} - \frac{f(U) - f(U-\Delta U)}{\Delta U}
\]

\[
= \frac{f(U+\Delta U)}{(\Delta U)^2} - \frac{2f(U)}{(\Delta U)^2} + \frac{f(U-\Delta U)}{(\Delta U)^2}
\]  

(3.3a)

\[
f_x^{(2)}(x,y) = \frac{f(X+\Delta X,Y)}{(\Delta X)^2} - \frac{2f(X,Y)}{(\Delta X)^2} + \frac{f(X-\Delta X,Y)}{(\Delta X)^2}
\]  

(3.3b)

\[
f_y^{(2)}(x,y) = \frac{f(X,Y+\Delta Y)}{(\Delta Y)^2} - \frac{2f(X,Y)}{(\Delta Y)^2} + \frac{f(X,Y-\Delta Y)}{(\Delta Y)^2}
\]  

(3.3c)

The substitutions for \( \Delta U \) in the derivative definition, Equations 3.1, make the first-order difference, Equations 3.2, unbiased. Otherwise a choice has to be made about the direction of the limiting value of \( \Delta U=0 \), which may be \( \Delta U \rightarrow 0^+ \) or \( \Delta U \rightarrow 0^- \). The second-order difference, Equations 3.3, is inherently unbiased.

Differencing operations are accomplished by digital filter matrix convolution. Digital filter convolution may use matrix operations to accomplish the Equations 3.1 through 3.3 differencing operations. Equations 3.2 and 3.3 are unbiased because the differencing operations are symmetric around the convolution window's central \( f(X,Y) \) grid post.
3.1.2 Matrix Convolution Formulation

Convolution can be thought of as a mathematical sampling procedure. Point spread sampling functions are convolved with functions to be sampled.

\[ h(s,t) = \text{point spread sampling function} \]
\[ f(x,y) = \text{function surface to be sampled} \]
\[ g(x,y) = \text{filtered surface of sample values} \]

\[
g(x,y) = \int \int f(x,y) h(x-s,y-t) \, ds \, dt
= \int \int f(x-s,y-t) h(s,t) \, ds \, dt \tag{3.4}
\]
\[
g(X_i, Y_j) = \sum_s \sum_t f(X_i-s, Y_j-t) \, h(s,t) \tag{3.5}
\]

Filtering occurs by convolving the point spread sampling function, \( h(s,t) \), on the function surface to be sampled, \( f(x,y) \), producing the filtered surface of sample values, \( g(x,y) \). Equation 3.4 is a continuous function convolution. The integrand is the product of two functions, \( f(x,y) \) and \( h(s,t) \), with the \( h(0,0) \) function placed at the \((x,y)\) position. The \( h(s,t) \) function is spread over the spatial domain's \( x \) and \( y \) dimensions by the \((s,t)\) parameters, respectively. Equation 3.5 is discrete function convolution.

Digital filtering is a special matrix operation. Equation 3.5 describes digital filter matrix convolution on
the grid lattice of the digital image. In that case \( F \), \( G \), and \( H \) are matrices whose elements are discrete values of the \( f \), \( g \), and \( h \) functions respectively. \( H \) is called the digital filter. Each resultant element of the \( G \) matrix is the summation of matrix element products between the \( F \) and \( H \) matrices, which we indicate by

\[
g(X_i, Y_j) = \frac{F' \ast H}{r,c} = \frac{3 \ast 3}{3,3}
\]

The convolution window's dimensions are specified as row by columns, \((r,c)\). \( H \) is a matrix of row by column dimensions. The \( F' \) matrix is a row by columns partition of the \( F \) matrix. In Equations 3.5 and 3.6, \( H \) is being convolved on \( F \). Each \( h(s,t) \) element of the \( H \) matrix will be multiplied by each corresponding \( f(X_i-s, Y_j-t) \) element of the \( F' \) matrix partition, where \( f(X_i-s, Y_j-t) \) is the image brightness function evaluated at each grid post's spatial \((X_i-s, Y_j-t)\) position. The \( f(X_i-s, Y_j-t) \) and \( h(s,t) \) matrix element products will be summed and then assigned to the \( g(X_i, Y_j) \) element of the \( G \) matrix. \( f(X_i, Y_j) \) is the convolution window's central post. Likewise, \( h(0,0) \) is the digital filter's central post.

The matrix formats of Equation 3.5 and 3.6 in the three by three digital filter are

\[
g(X_i, Y_j) = \frac{F' \ast H}{3,3} = \frac{3 \ast 3}{3,3}
\]
A directional ambiguity is inherent in the mathematical definition of convolution because of the 180° rotation of the point spread sampling function. A switch of direction will be introduced deliberately into the $H$ matrix while using matrix convolution to estimate derivatives by differences. In Equation 3.5 substitutions are made for the $H$ matrix element's positions where $h(s,t)=h(-s,-t)$ so that the matrix convolution's direction will match the difference's direction. An alternative matrix form, Matrix 3.7d, is produced. One finds that rotation invariant (isotropic) mathematical operations that are mimicked by digital filter matrix convolution are unaffected by the directional ambiguity of $H$, anyway.
Unbiased differencing operations result in a symmetric digital filter, $H$, where the digital filter's central element is $h(0,0)$. Matrix convolution is symmetric when the $(2m+1)$ by $(2n+1)$ dimensioned $H$ matrix elements are $h(s,t)$, where $s=(-m,\ldots,-1,0,1,\ldots,m)$ and $t=(-n,\ldots,-1,0,1,\ldots,n)$.

Developing the digital filter matrix convolution format (Equations 3.5 through 3.7) for the second-order differences (Equations 3.3) in the orthogonal $x$ and $y$ directions, we obtain the following:

$$
\begin{align*}
\mathbf{H}_x &= \begin{bmatrix}
0 & (\Delta X)^{-2} & 0 \\
0 & -2(\Delta X)^{-2} & 0 \\
0 & (\Delta X)^{-2} & 0
\end{bmatrix} \\
\mathbf{H}_y &= \begin{bmatrix}
(\Delta Y)^{-2} & -2(\Delta Y)^{-2} & (\Delta Y)^{-2} \\
0 & 0 & 0
\end{bmatrix}
\end{align*}
$$

The $H_x$ matrix's middle column vector is identical to the $H_y$ matrix's middle row vector.
3.1.3 Grid Post Spacing Weight Matrix

The grid post spacing affects the digital filter’s ability to approximate the derivative using discrete difference step intervals. It may be desirable to make the actual point spread function’s digital filter and the grid post spacing separable functions, because the post spacing will affect the slope calculations. It is possible to weight each \( h(s,t) \) element of the \( H \) matrix according to grid post spacing distance.

Separating the grid post spacing from the point spread function will transform the \( H \) matrix to a weighted \( H' \) matrix. The derivative’s denominator becomes an inverse distance that forms the elements of the weight matrix, \( W \). The derivative’s numerator becomes the elements of the identity digital filter, \( H' \). Let \( H \) be a special matrix of element products ('o') between the two matrices defined by

\[
H = H' \circ W
\]  

(3.9)

where \( h(s,t) = h'(s,t) w(s,t) \) are matrix element products. The matrix element product operation ('o') is different than the matrix convolution operation ('*'), because the matrix element products are not summed. Using the 'o' matrix element product and rewriting the matrices defined in Equation 3.8, we obtain
\[
\begin{align*}
\frac{H_{x}}{3,3} &= \frac{H'_{x}}{3,3} \circ \frac{W}{3,3} \\
&= \begin{bmatrix}
0 & 1 & 0 \\
0 & -2 & 0 \\
0 & 1 & 0
\end{bmatrix} \
&\quad \begin{bmatrix}
0 & (\Delta x)^{-2} & 0 \\
0 & (\Delta x)^{-2} & 0 \\
0 & (\Delta x)^{-2} & 0
\end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\frac{H_{y}}{3,3} &= \frac{H'_{y}}{3,3} \circ \frac{W}{3,3} \\
&= \begin{bmatrix}
0 & 0 & 0 \\
1 & -2 & 1 \\
0 & 0 & 0
\end{bmatrix} \
&\quad \begin{bmatrix}
0 & 0 & 0 \\
(\Delta y)^{-2} & (\Delta y)^{-2} & (\Delta y)^{-2} \\
0 & 0 & 0
\end{bmatrix}
\end{align*}
\]

One may notice that the \(H\) matrix's directional ambiguity will not affect the digital filter's form because \(h(s,t)=h(-s,-t)\) in Equations 3.10.

3.2 Definition Of An Edge

Edges are now defined by the mathematical operations that are used to determine them. Edges are also described in terms of their geometric qualities.

3.2.1 One-Dimensional Edge

Edges are points on one-dimensional profiles. The simplest and most common edge is the step edge. The study of the edge point on a continuous one-dimensional profile will give insight into digital filtering of a
two-dimensional image using matrix convolution.

3.2.1.1 Digital Step Edge Filtering

Figure 3.1 illustrates the one-dimensional digital step edge. The edge is formed by a grid post value difference step of magnitude |b-a| between posts three and four. All other grid post value difference steps equal zero that is the expected grid post value difference step along the profile.

One-dimensional digital filters are vectors. The vectors ignore the y and t dimension of Equation 3.5. The digital filter convolution is

\[ g(U_i) = \sum_s f(U_i-s) h(s) \]
\[ = f(U_i-1)h(-1) + f(U_i)h(0) + f(U_i+1)h(1) \]

The \( i \)th-order difference digital filter is the
\( H_i=[h(-1),h(0),h(1)] \) vector. The digital filters are

\[
\begin{align*}
\mathbf{H}_0^{1,3} &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} = f(u) \\
\mathbf{H}_1^{1,3} &= \mathbf{H}'_1 \circ \mathbf{W}_1 \\
&= \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \circ \begin{bmatrix} 1/2 & 0 & 1/2 \end{bmatrix} \\
&= \begin{bmatrix} -1/2 & 0 & 1/2 \end{bmatrix}
\end{align*}
\]
Figure 3.1: Digital Step Edge
where $\Delta U=1$ is assumed to be unit distance grid post spacing in the $y$ direction along the digital profile. The first-order difference has to occur over double post spacing, $2\Delta U$, for the digital filter to be unbiased. Otherwise a choice must be made between $H_1=[-1,1,0]$ or $H_1=[0,-1,1]$, which shifts the first-order difference to the negative or positive side of the convolution window's central post.

The filtered step edge is then computed to be

\[
G_1 = F' * H_1 = \begin{bmatrix} a & a & a & b & b & b \end{bmatrix} \ast \begin{bmatrix} -1/2 & 0 & 1/2 \end{bmatrix} = \begin{bmatrix} ? & 0 & 0 & (b-a)/2 & (b-a)/2 & 0 & 0 & ? \end{bmatrix} \tag{3.15}
\]

\[
G_2 = F' * H_2 = \begin{bmatrix} a & a & a & b & b & b \end{bmatrix} \ast \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} = \begin{bmatrix} ? & 0 & 0 & (b-a) & (a-b) & 0 & 0 & ? \end{bmatrix} \tag{3.16}
\]

Figure 3.2 displays the convolution results. The '? elements of each vector are indeterminant unless the ends of $F'$ are extended to accommodate the span of the digital filter. On the step edge, the magnitudes of the non-zero
Figure 3.2: Filtered Digital Step Edge
elements for $G_2$ are $2\Delta U$ times greater than the magnitudes of the non-zero elements of $G_1$.

3.2.1.2 Continuous Ramp Filtering

The presence of blur and noise turns step edges into noisy ramps. The ramp smoothed to remove noise resembles the arctangent function's profile.

\[
f(0)(u) = v \tan^{-1}(hu)
\]

\[
f(1)(u) = \frac{vh}{1 + h^2u^2}
\]

\[
[f(1)(u)]^2 = \frac{v^2h^2}{(1 + h^2u^2)^2}
\]

\[
f(2)(u) = (vh)(-1)(2uh^2)/(1 + h^2u^2)^2
\]

\[= (-2uhh^2v^2/v)/(1 + h^2u^2)^2
\]

\[= (-2uh/v)[f(1)(u)]^2
\]

\[
f(3)(u) = (-2h/v)[f(1)(u)]^2 + \]
\[(-2uh/v)(2)[f(1)(u)][f(2)(u)]
\]
\[= (-2h/v)[f(1)(u)]^2 +
\]
\[(-2uh/v)(2)[f(1)(u)](-2uh/v)[f(1)(u)]^2
\]
\[= (-2h/v)[f(1)(u)]^2 +
\]
\[(-2uh/v)(2)(-2uh/v)[f(1)(u)]^3
\]
\[= (-2h/v)[f(1)(u)]^2(1 - (4u^2h/v)[f(1)(u)])
\]
\[= (-2h/v)[f(1)(u)]^2(1 - (4u^2h^2)/(1 + h^2u^2))
\]
where \( u = x - x_0 \) is the distance from the center of the ramp, \( x_0 \), and \( f(u) \) is brightness at position \( u \).

The center of the ramp, \( x_0 \), is the arctangent profile's inflection point. The developed arctangent function's derivatives include horizontal and vertical scalars, \( h \) and \( v \) respectively. Equation 3.17d equals zero, \( f^{(3)}(u) = 0 \), can be used to find critical edge points on the arctangent function's profile. Figure 3.3 plots the zero-order through second-order derivatives for the arctangent function that represents the continuous ramp.

When examining the first-order and second-order derivatives of the edge's continuous profile formed by the arctangent function, the edge detection utilities of each derivative become apparent. Figure 3.3 illustrates that the first-order derivative has a local maxima at the middle of the ramp, \( x_0 \). The second-order derivative has a local maxima or a local minima on the ramp's shoulders. The magnitude of the second-order derivative is less than the magnitude of the first-order derivative. The second-order derivative includes a zero crossing between the maxima and minima at the middle of the ramp, \( x_0 \), at the arctangent function profile's inflection point.

Discrepancies between discrete differences versus continuous derivatives become apparent when the arctangent function's continuous profile is digitized and then filtered by using the digital filters in Equations 3.15 and 3.16.
Figure 3.3: Continuous Ramp Arctangent Derivatives
Figure 3.4 illustrates the discrepancies between difference versus derivatives.

The first-order difference’s ability to mimic the first-order derivative is considerably hindered because of the restriction that the filter must be unbiased, causing $2\Delta U$ to be used when estimating derivatives by slopes. The smaller $\Delta U$ for slope calculations using the second-order difference filter is possible because the filter is inherently unbiased. A smaller $\Delta U$ should be preferable when the profile is unpredictable because $\Delta f(U)/\Delta U$ only approaches $\delta f(u)/\delta u$ as $\Delta U$ decreases.

### 3.2.2 Two-Dimensional Edge

An edge is a line on a two-dimensional surface that may represent the image brightness surface. Gradients on the mathematical surface, $f(x,y)$, are approximated by grid post value difference steps between adjacent grid post values in two orthogonal directions.

### 3.2.2.1 Surfaces Of Rotation

A mathematical surface of rotation can be designed that contains edges with constant circular curvature if the function representing an edge profile, $f(R)$, is put on all radials from a single radix, $(X_0,Y_0)$, in the spatial domain. Let the surface of rotation formulations be
Figure 3.4: Arctangent Derivatives Versus Differences
\[
x' = X - X_0 \\
y' = Y - Y_0 \\
R = \sqrt{x'^2 + y'^2}
\]

Then
\[
f(x', y') = f(R) \tag{3.18}
\]

Curvature is a differential geometry concept that is used to describe the shape of a line. Kresig (1983) and Larson and Hostetler (1979) describe the concept of curvature. Curvature describes a tangent circle whose differential vectors are equal to the curve's differential vectors at the point where the curve's curvature is being calculated. A circle has constant curvature of \( \kappa = 1/R \), where \( R \) is the circle's radius.

Figure 3.5 illustrates the surface formed by rotating the arctangent function's profile. The edge at a point on the radial profile becomes a circular line that has a constant curvature on the two-dimensional surface. An edge of constant curvature (\( \kappa = 1/R \)) will have continuously varying orientation with respect to the grid lattice used to sample the two-dimensional surface.

3.2.2.2 Edge Geometry

The edge curvature (shape) combined with the radius of curvature direction (edge line orientation) is sufficient to describe the geometry of the edge at a particular point
Figure 3.5: Radial Arctangent Profile's Surface of Rotation
that is on the edge's line. The direction of the radius of curvature vector, that we shall label as \( n \), is determined from the vector cross product of the tangent vectors in the \( x \) and \( y \) directions. The direction of \( n \) is the edge line's orientation at that \((x,y)\) point. Let

\[
z = f(x,y)
\]

Then direction of the tangents in the \( x \) and \( y \) direction are

\[
\begin{align*}
\tau_x &= \tan^{-1}(\partial z/\partial x) \\
\tau_y &= \tan^{-1}(\partial z/\partial y)
\end{align*}
\]

and the radius of curvature vector is

\[
n = [i(\cos \tau_x) + i(0) + k(\sin \tau_x)]
\]

\((X)\); vector cross-product

\[
= [i(0) + i(\cos \tau_y) + k(\sin \tau_y)]
\]

\[
- i(\sin \tau_x)(\cos \tau_y) - i(\cos \tau_x)(\sin \tau_y) + k(\cos \tau_x)(\cos \tau_y)
\]

The edge's orientation in the \((x,y)\) spatial domain is labeled as \( \theta \). The edge's orientation is calculated by ignoring the \( k \) element of the \( n \) vector, and then taking the arctangent of the ratio of the \( i \) and \( j \) elements of the \( n \) vector, so that
\[ \theta = \tan^{-1} \left[ \frac{\cos \tau_x (\sin \tau_y)}{\sin \tau_x (\cos \tau_y)} \right] \]

\[ = \tan^{-1} \left[ (\cotan \tau_x)(\tan \tau_y) \right] \]

\[ = \tan^{-1} \left[ (\partial x/\partial z)(\partial z/\partial y) \right] \]

\[ = \tan^{-1} \left[ (\partial z/\partial y) / (\partial z/\partial x) \right] \]

(3.20)

The shape and orientation of the edge may be estimated in a digital image by substituting the discrete differences of \((\Delta x, \Delta y, \Delta z)\) for \((\partial x/\partial z, \partial y, \partial z/\partial x)\), respectively.

The edge sharpness may be described by the derivatives along the one-dimensional brightness profile corresponding to the direction, \(\theta\), of the perpendicular radius of curvature vector, \(n\). If the edge is being described by the arctangent function's profile, then the \(v/h\) ratio from Equations 3.17 can describe the edge's sharpness.

3.3 Laplacian Discrete Difference Method

The Laplacian is a linear function of second-order derivatives that has the property of rotation invariance. The Laplacian value is insensitive to the orientation of the surface with respect to the coordinate system. Those properties will justify the use of the digital Laplacian to compare grid lattice effects on filter convolution.
3.3.1 Laplacian’s Mathematical Definition

Rotation invariant derivative operations are desirable to detect edge features. A filtered edge of constant shape and sharpness should produce a constant value after differentiation, independent of the edge’s orientation in the coordinate system that is called rotation invariance. Isotropic differentiation operations are rotation invariant. Rosenfeld and Kak (1982) list the following two properties of isotropic differentiation:

1. An isotropic linear derivative operator can involve derivatives of even order.

2. In an arbitrary isotropic derivative operator, derivatives of odd order can occur only raised to even powers.

The exponentiated derivative, \([f^{(i)}(x,y)]^j\), is only isotropic when \(i\) multiplied by \(j\) equals an even number. The mathematical Laplacian is an isotropic differentiation operation that is given by Equation 3.21.

\[
\chi[f(x,y)] = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}
= f_x^{(2)}(x,y) + f_y^{(2)}(x,y) 
\]

(3.21)

The mathematical Laplacian adheres to property number one that makes it a rotation invariant derivative operation.
Appendix A.1 verifies the rotation invariance of the Laplacian.

If we apply the Laplacian to the arctangent profile's surface of rotation, defined by Equations 3.17 and 3.18, then a constant value along the circular edge line occurs.

\[
\zeta[f(x',y')] = \frac{\partial^2 f(x',y')}{\partial x'^2} + \frac{\partial^2 f(x',y')}{\partial y'^2}
\]

\[
= f(2)(R) - f(1)(R)/R
\]

\[
= \frac{\partial^2 f(R)}{\partial R^2} - \frac{\partial f(R)}{\partial R} / R \quad (3.22)
\]

The Laplacian is a derivative operator that will give a circular edge line a constant value because \( \zeta[f(x',y')] \) equals a function of \( R \) only.

3.3.2 Equivalent Digital Laplacian Filters

When differencing with digital filter matrix convolution an attempt is made to maintain the rotation invariant properties of the derivative, otherwise edge detection attempts are complicated by the edge’s orientation on the image plane.

The 4-neighbor grid’s digital Laplacian filter is formed by using second-order differences in the orthogonal \( \Delta X \) and \( \Delta Y \) directions to estimate the second-order derivatives in each respective direction.
The digital Laplacian filter is developed from Equations 3.2 through 3.10. Equation 3.23 is the digital Laplacian convolution at the grid post position, \((X_i,Y_j)\).

\[ z[f(x,y)] = g(X_i,Y_j) \]  

\[ = \frac{F'}{3,3} * \frac{H_x}{3,3} + \frac{F'}{3,3} * \frac{H_y}{3,3} \]

\[ = \frac{F'}{3,3} * \left( \frac{H_x}{3,3} + \frac{H_y}{3,3} \right) \]

\[ = \frac{F'}{3,3} \begin{bmatrix} 0 & (\Delta X)^{-2} & 0 \\ (\Delta Y)^{-2} & -2(\Delta X)^{-2} + 2(\Delta Y)^{-2} & (\Delta Y)^{-2} \\ 0 & (\Delta X)^{-2} & 0 \end{bmatrix} \]

\[ = \frac{F'}{3,3} * \frac{H_4}{3,3} \]

When there is unit distance grid post spacing in both directions, \(\Delta X = \Delta Y = 1\), the 4-neighbor grid's digital Laplacian filter becomes an identity filter, \(H_4 = H'_4\), that is shown in Equation 3.24.

\[ H'_4 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \]  

(3.24)
Each n-neighbor grid’s digital Laplacian filter was designed by identifying a salient property of the 4-neighbor grid’s identity digital Laplacian filter. In the 4-neighbor grid’s identity digital Laplacian filter, the matrix convolution produces the sum of differences between the central post and the neighboring grid post values in the preferential directions of x and y.

\[ z[f(x,y)] = g(x_i, y_j) \]
\[ = [ f(x_i, y_{j-1}) + f(x_{i-1}, y_j) + f(x_{i+1}, y_j) + f(x_i, y_{j+1}) ] - 4f(x_i, y_j) \]
\[ = f(x_i, y_{j-1}) - f(x_i, y_j) + f(x_{i-1}, y_j) - f(x_i, y_j) + f(x_{i+1}, y_j) - f(x_i, y_j) + f(x_i, y_{j+1}) - f(x_i, y_j) \]
\[ (3.25) \]

In a digital Laplacian filter, that sum of differences property produces a filter whose matrix elements sum equals zero.

\[ \sum_t \sum_s h(s,t) = 0 \]
\[ (3.26) \]

The digital Laplacian filters for each grid lattice were designed so that the matrix elements sum equals zero.
The spatial positions of the adjacent grid posts in the convolution window have already been presented in Table 2.2. Table 3.1 presents the identity digital Laplacian filters, \( H' \), and their respective weight matrices, \( W \), in terms of grid post spacing distances in the orthogonal \( \Delta X \) and \( \Delta Y \) directions for the 4-neighbor, 6-neighbor, and 8-neighbor grid lattice.

The equivalent digital Laplacian filters are developed by assuming unit distance grid post spacing, \( \Delta X = \Delta Y = 1 \), for the 4-neighbor and 8-neighbor weight matrices, \( W_4 \) and \( W_8 \). And the \( H'_6 \) identity digital Laplacian filter was weighted with grid post spacing density equivalent to \( H_8 \).

\[
H_4 = H'_4 \circ W_4
\]

\[
= \begin{bmatrix}
0 & 1 & 0 \\
1 & -4 & 1 \\
0 & 1 & 0
\end{bmatrix} \circ \begin{bmatrix}
0 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 0
\end{bmatrix}
\]

\[
= \begin{bmatrix}
0 & 1 & 0 \\
1 & -4 & 1 \\
0 & 1 & 0
\end{bmatrix}
\]

\[(3.27)\]
Table 3.1a: 4-Neighbor Digital Laplacian Filter Matrices

Filter Matrix

$$\mathbf{H}_4^{1,2} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Weight Matrix

$$\mathbf{W}_4^{1,2} = \begin{bmatrix} 0 & (\Delta X)^{-2} & 0 \\ (\Delta Y)^{-2} & w & (\Delta Y)^{-2} \\ 0 & (\Delta X)^{-2} & 0 \end{bmatrix}$$

$$w = \frac{[2(\Delta X)^{-2} + 2(\Delta Y)^{-2}]}{4}$$
Table 3.1b: 6-Neighbor Digital Laplacian Filter Matrices

**Filter Matrix**

\[ \begin{bmatrix} 1 & 1 & 1 \\ 0 & -6 & 0 \\ 1 & 1 & 1 \end{bmatrix} \]

**Weight Matrix**

\[ \begin{bmatrix} (\Delta')^{-2} & (\Delta X)^{-2} & (\Delta')^{-2} \\ 0 & w & 0 \\ (\Delta')^{-2} & (\Delta X)^{-2} & (\Delta')^{-2} \end{bmatrix} \]

\[ \Delta' = \left( \left(\frac{\Delta X}{2}\right)^2 + (\Delta Y)^2 \right)^{1/2} \]

\[ w = \left[ 2(\Delta X)^{-2} + 4(\Delta')^{-2} \right] / 6 \]
Table 3.1c: 8-Neighbor Digital Laplacian Filter Matrices

**Filter Matrix**

\[ \mathbf{H}_{8}^{3,3} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \]

**Weight Matrix**

\[ \mathbf{W}_{8}^{3,3} = \begin{bmatrix} (\Delta')^{-2} & (\Delta X)^{-2} & (\Delta')^{-2} \\ (\Delta Y)^{-2} & w & (\Delta Y)^{-2} \\ (\Delta')^{-2} & (\Delta X)^{-2} & (\Delta')^{-2} \end{bmatrix} \]

\[ \Delta' = \left( (\Delta X)^2 + (\Delta Y)^2 \right)^{1/2} \]

\[ w = \left[ 2(\Delta X)^{-2} + 2(\Delta Y)^{-2} + 4(\Delta')^{-2} \right] / 8 \]
\[ H_6 = H'_6 \circ W_6 \]

\[
= \begin{bmatrix}
1 & 1 & 1 \\
x & -6 & x \\
1 & 1 & 1
\end{bmatrix}
\times
\begin{bmatrix}
(3/4)^{1/2} & (3/4)^{1/2} & (3/4)^{1/2} \\
x & (3/4)^{1/2} & x \\
(3/4)^{1/2} & (3/4)^{1/2} & (3/4)^{1/2}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
1/2 & 1/2 & 1/2 \\
x & -3 & x \\
1/2 & 1/2 & 1/2
\end{bmatrix}
\]

(3.28)

\[ H_8 = H'_8 \circ W_8 \]

\[
= \begin{bmatrix}
1 & 1 & 1 \\
1 & -8 & 1 \\
1 & 1 & 1
\end{bmatrix}
\times
\begin{bmatrix}
1/2 & 1 & 1/2 \\
1 & 3/4 & 1 \\
1/2 & 1 & 1/2
\end{bmatrix}
\]

\[
= \begin{bmatrix}
1/2 & 1 & 1/2 \\
1 & -6 & 1 \\
1/2 & 1 & 1/2
\end{bmatrix}
\]

(3.29)

The 'x' matrix elements are unused thus making \( H_6 \) a sparse matrix. The \( H_6 \) matrix's diagonal directions correspond to the 6-neighbor grid lattice elementary window's \( u \) and \( w \) preferential directions (see Figure 2.1).
All three weighted digital Laplacian filters \((H_4, H_6,\) and \(H_8\)) still maintain the matrix elements sum equals zero property. That resulted because the weight matrices were designed to account for different grid post spacing without removing the matrix elements sum equals zero property.

### 3.3.3 Other Isotropic Derivative Operations

The sum of the first-order derivatives squared will produce an isotropic derivative operation that is called the squared magnitude of the gradient.

\[
M^2[f(x,y)] = [\partial f(x,y)/\partial x]^2 + [\partial f(x,y)/\partial y]^2
\] (3.31)

However, the need to square the differences results in cross product terms between grid posts.

\[
[f(X_{i+1},Y_j)-f(X_{i-1},Y_j)]^2 = [f(X_{i+1},Y_j)]^2 + [f(X_{i-1},Y_j)]^2 - 2[f(i+1,j)][f(i-1,j)]
\] (3.32)

It is impossible to incorporate unbiased cross product terms between grid posts into digital filter matrix convolution. Squaring operations and cross product terms are computationally expensive. The digital Laplacian filter matrix convolution uses computationally inexpensive multiplication and summation operations.
Reduced grid post spacing in the differencing equations will detect high frequency changes in grid post values. The grid post spacing distance will determine the digital filter's frequency pass. Unbiased first-order differences occur over twice the grid post spacing distance in respective preferential directions. Therefore, first-order difference filters will detect lower frequencies in the 3x3 convolution window than the second-order differences, which are actually occurring over just the grid post spacing distance in each preferential direction. Thus, unbiased first-order difference filters should be considered low frequency pass filters. Unbiased second-order difference filters should be considered high frequency pass filters.

3.4 Summary

All of this chapter's formulations have been developed to design an experiment for quantifying digital filtering grid lattice effects. The development of an arctangent function's surface of rotation creates a mathematical surface where a Laplacian can be calculated, and digital Laplacian filter convolution results can be compared to the mathematical Laplacian. Each edge's shape, orientation, and sharpness can be controlled on the mathematical surface. Finally, the mathematical development of digital filter matrix convolution, and the mathematical surface developed to test grid lattice effects, are flexible design tools for
developing any digital filters.
4. COMPARISON OF DIGITAL LAPLACIAN FILTERS

The formulations from previous chapters are used to quantify digital filtering grid lattice effects. An experiment is conducted to show a fair comparison between n-neighbor digital filters. A fair comparison means that all sources of digital filtering error have been removed except edge line orientation and grid lattice preferential directions. The experimental results suggest the preferable grid lattice while assuming that digital Laplacian filter convolution is a desirable edge detection operator.

4.1 Generation Of Test Data Set

Mathematical Laplacians were calculated from derivatives on the continuous two-dimensional mathematical surface of rotation. An edge point on a one-dimensional profile, \( f(U) \), became a curved edge line, \( f(R) \), when the radial profile swept out the surface of rotation.

Each circular edge line had a constant curvature, \( \kappa = 1/R \), and variable edge orientation, \( \theta \), on a mathematical surface of rotation. Figure 4.1 illustrates a circular edge line of constant curvature and variable orientation, on a mathematical surface of rotation.

Equation 3.22 showed that the mathematical Laplacian will have a constant value along a surface of rotation’s circular edge lines, because \( \ell[f(x', y')] \) only equals a
Figure 4.1: Circular Edge Line Sample Post Orientation
function of the edge line's radius of curvature, 
\[ R = \left[ x'^2 + y'^2 \right]^{1/2}. \]
In fact, the difference between the second-order derivative on the one-dimensional mathematical profile, \( f^{(2)}(U) \), and the mathematical Laplacian on the two-dimensional mathematical surface of rotation is proportional to the circular edge line's curvature, \( \kappa = \frac{1}{R} \), and the mathematical surface of rotation's first-order differential.

\[
[f^{(2)}(U)] - [f^{(2)}(R)] = \frac{[f^{(1)}(R)]}{R} = \kappa [f^{(1)}(R)] \tag{4.1}
\]

The edge detection utility of the Laplacian was described using the Section 3.2.1.2 examplar arctangent function profile, representing the simple step edge's ramp. The detection of that simple edge had three critical points when the second-order derivative of the one-dimensional arctangent function was examined. The critical points included a maximum, minimum, and an inflection. The critical points were calculated by solving for \( f^{(3)}(u) = 0 \), where \( u = x - x_o \) and \( x_o \) is the middle of edge ramp. The zero roots of the third-order derivative to \( f^{(0)}(u) = (v)\tan^{-1}(hu) \) are \( u = 0 \) and \( u = \pm [(3^{1/2})h]^{-1} \).

The critical points along the arctangent function's one-dimensional profile will be picked as critical radial values for circular edge lines on the arctangent function.
radial profile’s surface of rotation. Thus, the n-neighbor digital Laplacian filter convolution results can be compared to the actual mathematical Laplacians along three circular edge lines that have constant mathematical Laplacian values. The three circular edge lines each will have the radius of \( R_{\text{max}} = R - [(3^{1/2})h]^{-1} \), \( R_{\text{mid}} = R \), and \( R_{\text{min}} = R + [(3^{1/2})h]^{-1} \). The radii’s subscript (\( \text{max} \), \( \text{mid} \), \( \text{min} \)) refer to critical values for the one-dimensional arctangent function’s second-order derivative.

Equivalent digital Laplacian filters were derived from the 4-neighbor grid lattice’s characteristics when there was unit distance grid post spacing in the orthogonal \( \Delta X \) and \( \Delta Y \) spatial directions, \( \Delta X = \Delta Y = 1 \).

A sample of mathematical Laplacian minus digital Laplacian filter matrix convolution results are collected along each circular edge line for each n-neighbor grid lattice. Sample posts are picked in regular arc increments along each circular edge line. A complete sample is produced of a particular circular edge line of constant shape and sharpness but with a variety of edge orientations.

The Table 2.2 displacement vector index arrays will be applied to the edge line’s sample post position when each grid lattice’s digital filter convolution occurs. The convolution window’s central grid post has the \( (\Delta X, \Delta Y) = (0, 0) \) displacement vector for all grid displacement vector index array matrices, so all of the digital filter’s central posts
will be in identical edge line sample post positions. The different \( \Delta X \) and \( \Delta Y \) values in each grid lattice’s displacement vector index array will put the posts that surround the convolution window’s central post into different positions, because the digital filters were designed for equal grid post sampling density between each grid lattice. The identical central post positions for the digital filters prevent the need to interpolate while comparing each grid lattice’s digital filter matrix convolution results at each edge line’s sample post. Figure 4.2 shows the convolution windows of the equal density 6-neighbor grid lattice versus the 8-neighbor grid lattice when they share the same sample post.

Software was designed to accommodate the experiment that quantifies digital filtering grid lattice effects. The software presented in Appendix B.2 produced the Appendix B.3 data files. Various software modules were developed to specify the continuous two-dimensional mathematical surface of rotation, plus specify digital filters and do digital filter matrix convolution on the mathematical test surface. Appendix B describes and presents the software that produced this thesis’s test results. The data files form the discrepancy curves that are presented in the next section.

The differences between the constant mathematical Laplacian along each circular edge line and the corresponding digital Laplacian filter convolution results
Figure 4.2:

6-Neighbor And 8-Neighbor Grid's Elementary Window
for each grid lattice will determine the preferred grid lattice.

4.2 Digital Laplacian Filtering Grid Lattice Results

Discrepancy curves compare the mathematical Laplacians minus digital Laplacians along circular edge lines that have a constant mathematical Laplacian on the mathematical surface of rotation.

The digital Laplacian will not have a constant value, because of variable directional phases between edge line orientation and grid lattice preferential directions. The 4-neighbor, 6-neighbor, 8-neighbor digital Laplacian filtering results are compared, while each filter is applied to particular circular edge lines ($R_{\text{max}}, R_{\text{mid}}, R_{\text{min}}$) with sample points having different edge orientations, $\theta$, on the mathematical surface of rotation that is formed by the arctangent function's radial profile.

Discrepancy curves are formed by the absolute value of the difference between the n-neighbor grid lattice digital Laplacian filtering results and the mathematical Laplacian, for a complete set of sample posts at edge orientations of zero through ninety degrees. There were no zero-crossings by any of the discrepancy curves, so the absolute value of the difference could be used to increase scale of the y-axis. The edge orientations will only vary from zero to ninety degrees because of symmetry of the grid lattice.
geometry versus edge geometries from $90^\circ$ to $360^\circ$.

The discrepancy curves represent a distribution of
digital filter matrix convolution error where the circular
edge line's orientation is the only random variable, which
we denote as $\theta$ degrees on each discrepancy curve's x-axis.
We denote the mathematical Laplacian as $z[f(R)]$, because it
has been verified that the Laplacian's value can be
expressed as a function of the radius from the circular edge
to mathematical surface of rotation's radix. And we have
verified that $z[f(x',y')] = z[f(R)]$ has a constant value along
that defined circular edge line, $f(R)$.

We shall see the digital Laplacian filtering results
will not be constant. The random variable will be the
edge's orientation, $\theta$, on the x-axis. Since $R$ is constant
on the circular edge line but the digital Laplacian is notconstant, we shall denote each n-neighbor grid lattice's
results as a variant function of the edge's orientation,
g($X_i,Y_j$) = $g_n(R \cos \theta, R \sin \theta)$. The dependent variable will be
the absolute value of the mathematical Laplacian minus the
digital Laplacian for each n-neighbor grid lattice, which we
denote as $\xi_n(R,\theta)$ on the y-axis.

$$\xi_n(R,\theta) = |z[f(R)] - g_n(R \cos \theta, R \sin \theta)| \quad (4.4)$$

Figure 4.3 displays the discrepancy curves of mathematical
Laplacian minus the digital Laplacian for three different
Figure 4.3a: $R_{\text{max}}$ Digital Laplacian Discrepancy Curves
Figure 4.3b: Rmid Digital Laplacian Discrepancy Curves
Figure 4.3c: $R_{\text{min}}$ Digital Laplacian Discrepancy Curves
circular edge lines. Each curve presents the error distribution for each grid lattice on a single circular edge line, where the radius of curvatures, $R$, are critical points of the second-order derivative along the Equation 3.17 arctangent function's radial profile that forms the mathematical surface of rotation. The discrepancy curves, $\xi_n(R, \theta)$, are described by each curve's periodicity, critical points, and amplitude. Table 4.1 describes the discrepancy curves on each circular edge line of $R_{\text{max}} = 25 - [3^{1/2}]^{-1}$, $R_{\text{mid}} = 25$, and $R_{\text{min}} = 25 + [3^{1/2}]^{-1}$. Where $R$ is specified in terms of unit distance grid post spacing, $\Delta X = \Delta Y = 1$.

Each discrepancy curve's periodicity shows directional phases between edge orientation and the digital filter's preferential directions. The directional phases determine the magnitude of digital filtering error when estimating the Laplacian. Discrepancy curve critical points occur where edge orientation corresponds to a grid lattice's preferential direction, or where the edge orientation is most out of phase with a grid lattice's preferential direction.

The n-neighbor digital filter's discrepancy curve amplitude shows digital filtering precision with respect to edge orientation. A digital Laplacian filter's precision increases as the discrepancy curve's amplitude decreases. That precision will reduce the necessary threshold band widths necessary to detect edges of unknown orientation.
Precision is desirable when thresholding digital filtering results to detect an edge. Threshold band widths can be narrower and therefore more selective while segmenting the image by digital filtering for edge detection.

Table 4.1 shows the discrepancy curve's amplitude from digital filter convolution along circular edge lines that have constant mathematical Laplacian value. Note that the 6-neighbor digital filtering discrepancy curves have significantly smaller amplitude and hence more precision. The 6-neighbor grid lattice's precision is significantly greater than the other grid lattice's precision on all three critical edge lines.

Table 4.1: Digital Laplacian Discrepancy Curve Characteristics

<table>
<thead>
<tr>
<th>Filter</th>
<th>Amplitude</th>
<th>Periodicity</th>
<th>Critical Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.089</td>
<td>0.013</td>
<td>0.095</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>π/2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0°, 45°, 90°</td>
</tr>
<tr>
<td>6</td>
<td>0.013</td>
<td>0.004</td>
<td>0.011</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>π/3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0°, 30°, 60°, 90°</td>
</tr>
<tr>
<td>8</td>
<td>0.024</td>
<td>0.007</td>
<td>0.033</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>π/2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0°, 45°, 90°</td>
</tr>
</tbody>
</table>

The discrepancy curve amplitudes for the circular edge lines of $R_{\text{max}}$ and $R_{\text{min}}$ are a smaller proportion of each edge line's mathematical Laplacian than for $R_{\text{mid}}$. The digital
Laplacian can still be used to identify that there is a zero-crossing between the two shoulders of an edge ramp.

The addition of extra preferential directions to the 4-neighbor digital Laplacian always reduced each discrepancy curve’s amplitude. The 4-neighbor grid discrepancy curve’s amplitude exceeded both of the 6-neighbor and 8-neighbor discrepancy curve amplitudes on each circular edge line of radius R. The 8-neighbor grid doubled the number of preferential directions over the 4-neighbor grid, the 8-neighbor grid’s discrepancy curve amplitude decreased (or precision increased) accordingly. But the 8-neighbor grid could not compensate for the unequal magnitude of the diagonal displacement vectors (a and b compared to x and y) to gain the precision of the 6-neighbor grid with all displacement vectors (u, v, w) of identical magnitude. The 6-neighbor grid’s discrepancy curve’s amplitude was the smallest of all grid lattices on each edge line.

The 4-neighbor and 6-neighbor grid lattice’s discrepancy curve periodicity equals each digital Laplacian filter’s angular resolution between preferential directions. The 8-neighbor grid lattice’s discrepancy curve periodicity does not match the digital Laplacian filter’s angular resolution between preferential directions. That is because the 8-neighbor grid lattice’s angular resolution between displacement vectors of equal magnitude actually is 90° anyway.
5. CONCLUSIONS AND RECOMMENDATIONS

The geometric effects on mathematical operations that filter the digital image have been described in detail. A method has been developed to model and quantify grid lattice preferential direction effects when digital filter matrix convolution occurs.

If thresholding is being utilized to detect an edge of unknown orientation (but of constant circular shape and sharpness) with digital filtering, a 6-neighbor grid lattice is desirable because of the precision in estimating differential operations within a discrete data set. Random edge orientation makes the increased angular resolution from preferential directions with identical grid post spacing distances, of the 6-neighbor grid lattice, desirable with digital filtering.

A scanner should be designed to produce 6-neighbor grid lattice digital imagery. Appendix C suggests the design modifications that might be necessary to modify conventional scanners to produce 6-neighbor grid lattice digital imagery. Then a real data set could be used to compare grid lattice effects on actual imagery, instead of ideal continuous mathematical surfaces.

The Appendix suggests other digital filters and matrix convolution operations that can be used to filter a digital image. Grid lattice geometry effects on those
alternative digital filters should be studied by using mathematical surfaces of rotation with various radial profiles. The thesis's mathematical formulations, combined with ideal continuous two-dimensional mathematical surfaces, should be used as tools to design digital filters.
6. References


A. DERIVATIONS

A.1 Isotropic Differentiation

Isotropic derivative operations are rotation invariant. Which means that rotation invariant operations are unaffected by rotation of the coordinate system. Rosenfeld & Kak (1982) present the properties of the isotropic derivative operations.

(1) An isotropic linear derivative operator can involve derivatives of even order.

(2) In an arbitrary isotropic derivative operator, derivatives of odd order can occur only raised to even powers.

\[ (A.1) \]

In other words, \([f^{(1)}(x,y)]^j\) (the exponentiated derivative) is only isotropic when \(i\) (the order of the derivative) multiplied by \(j\) (the exponent) equals an even number \((2, 4, 6, \ldots)\).

An example of property one is the Laplacian:

\[ \nabla[f(x,y)] = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \quad (A.2) \]
An example of property two is the squared magnitude of the gradient.

\[ M^2 = \left[ \frac{\partial f(x,y)}{\partial x} \right]^2 + \left[ \frac{\partial f(x,y)}{\partial y} \right]^2 \] (A.3)

The implication of isotropic derivative operations in edge detection is that an edge will not change its value as its orientation on a continuous image surface changes.

Consider the \( f(x,y) \) unrotated coordinates and the \( f(x',y') \) coordinates rotated by angle \( \theta \).

\[
\begin{bmatrix}
  x \\
  y
\end{bmatrix}
= \begin{bmatrix}
  \cos(\theta) & -\sin(\theta) \\
  \sin(\theta) & \cos(\theta)
\end{bmatrix}
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix}
= \begin{bmatrix}
  \cos(\theta) & \sin(\theta) \\
  -\sin(\theta) & \cos(\theta)
\end{bmatrix}
\begin{bmatrix}
  x \\
  y
\end{bmatrix}
\]

That means

\[
x = x' \cos(\theta) - y' \sin(\theta)
\]
\[
y = x' \sin(\theta) + y' \cos(\theta)
\]
\[
x' = x \cos(\theta) + y \sin(\theta)
\]
\[
y' = -x \sin(\theta) + y \cos(\theta)
\] (A.4)

That produces the equalities of the partial derivatives
\[ \frac{\partial x}{\partial x'} = \frac{\partial y}{\partial y'} = \frac{\partial x'}{\partial x} = \frac{\partial y'}{\partial y} = \cos(\theta) \]
\[ \frac{\partial x}{\partial y'} = \frac{\partial y'}{\partial x} = -\sin(\theta) \]
\[ \frac{\partial y}{\partial x'} = \frac{\partial x'}{\partial y} = \sin(\theta) \]  
(A.5)

By the chain rule and substitution, the first-order derivatives are

\[ f = f(x, y) \]
\[ f' = f(x', y') \]

\[ \frac{\partial f}{\partial x'} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial x'} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial x'} \]
\[ = \frac{\partial f}{\partial x} \cos(\theta) + \frac{\partial f}{\partial y} \sin(\theta) \]

\[ \frac{\partial f}{\partial y'} = \frac{\partial f}{\partial y} \frac{\partial y}{\partial y'} + \frac{\partial f}{\partial x} \frac{\partial x}{\partial y'} \]
\[ = \frac{\partial f}{\partial y} \cos(\theta) - \frac{\partial f}{\partial x} \sin(\theta) \]

\[ \frac{\partial f}{\partial x} = \frac{\partial f'}{\partial x'} \frac{\partial x'}{\partial x} + \frac{\partial f'}{\partial y'} \frac{\partial y'}{\partial x} \]
\[ = \frac{\partial f'}{\partial x'} \cos(\theta) - \frac{\partial f'}{\partial y'} \sin(\theta) \]
The Laplacian, \( \nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \), is an isotropic derivative operation according to the A.1 properties. Therefore the operation is rotation invariant. The Laplacian’s rotation invariance can be verified by deriving the equality \( \nabla^2 [f(x,y)] = \nabla^2 [f(x',y')] \), using the chain rule between the rotated coordinate systems.

\[
\begin{align*}
\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} &= \frac{\partial}{\partial x'} \left( \frac{\partial f}{\partial x} \right) + \frac{\partial}{\partial y'} \left( \frac{\partial f}{\partial y} \right) \\
&= \frac{\partial}{\partial x'} \left( \frac{\partial f}{\partial x} \right) \cos(\theta) + \frac{\partial}{\partial y'} \left( \frac{\partial f}{\partial y} \right) \sin(\theta)
\end{align*}
\]
that verifies the equality of the Laplacian before and after the rotation of the coordinate system.
A.2 Digital Filters

Differences are used to approximate derivatives in a discrete data set. The use of differences to approximate derivatives is justified because of the derivative's definition.

\[
\frac{\delta f(u)}{\delta u} = \lim_{\Delta U \to 0} \frac{f(U + \Delta U) - f(U)}{\Delta U}
\]  
(A.13a)

\[
\frac{\delta^2 f(u)}{\delta u^2} = \lim_{\Delta U \to 0} \left[ \frac{f(U + \Delta U) - f(U)}{\Delta U} \frac{f(U) - f(U - \Delta U)}{\Delta U} \right] / \Delta U
\]

\[= \lim_{\Delta U \to 0} \frac{f(U + \Delta U)}{(\Delta U)^2} + \frac{f(U - \Delta U)}{(\Delta U)^2} - \frac{2 f(U)}{(\Delta U)^2}
\]  
(A.13b)

The second-order differences are used to approximate the Laplacian of a continuous function.

\[
z[f(x, y)] = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}
\]

\[= f_x^2(x, y) + f_y^2(x, y)
\]

\[= \lim_{\Delta X \to 0} \frac{f(X + \Delta X, Y) - f(X, Y)}{(\Delta X)^2} - \frac{f(X, Y) - f(X - \Delta X, Y)}{(\Delta X)^2} + \]

\[\lim_{\Delta Y \to 0} \frac{f(X, Y + \Delta Y) - f(X, Y)}{(\Delta Y)^2} - \frac{f(X, Y) - f(X, Y - \Delta Y)}{(\Delta Y)^2}
\]
\[ \lim_{\Delta X \to 0} \frac{f(x+\Delta x, y) - f(x, y)}{(\Delta x)^2} + \frac{f(x-\Delta x, y) - f(x, y)}{(\Delta x)^2} - \frac{2f(x, y)}{(\Delta x)^2} + \]

\[ \lim_{\Delta Y \to 0} \frac{f(x, y+\Delta y) - f(x, y)}{(\Delta y)^2} + \frac{f(x, y-\Delta y) - f(x, y)}{(\Delta y)^2} - \frac{2f(x, y)}{(\Delta y)^2} \]

\[ = \frac{f(x+\Delta x, y)}{(\Delta x)^2} + \frac{f(x-\Delta x, y)}{(\Delta x)^2} - \frac{2f(x, y)}{(\Delta x)^2} + \]

\[ + \frac{f(x, y+\Delta y)}{(\Delta y)^2} + \frac{f(x, y-\Delta y) - 2f(x, y)}{(\Delta y)^2} \]

\[ = f_x^2(x, y) + f_y^2(x, y) \] (A.14)

The difference formulas, \( f^{(1)} \), are formed by dropping the limit, \( \Delta U \to 0 \), from the derivative formulas, \( f^{(i)} \).

The denominators of the difference formulas are located in the weight matrix, \( W \), while the numerators are located in the identity matrix, \( I \). The discrete image is located in the \( F \) matrix. Matrix convolution of the equation \( \sum_s f(x_i-s, y_j-t)h(s, t) \) is used to approximate derivatives by differences. All of those convolution matrix design criterion are used to create the digital filter convolution matrices utilized in this thesis. The \( H_1 \) and \( H_2 \) vectors are used to compare first-order differencing and second-order differencing, respectively.

An aberration to equation A.13a appears to cause the first order difference 3x1 filter's central post, \( H_1(0,0) \),
to be coincident with the convolution window's sample post where the first-order difference is being assigned, to cause an unbiased filter.

\[
\frac{\delta f(u)}{\delta u} = \text{limit} \quad \frac{f(U + \Delta U) - f(U - \Delta U)}{2\Delta U} \quad (A.15)
\]

Formula A.15 allows the difference to be taken in a symmetric neighborhood of the matrix convolution's central post, formula A.13a does not. Equation A.15 in matrix convolution form is

\[
\begin{bmatrix}
H_{1,3} \\
H'_{1,3} \\
W_{1,3}
\end{bmatrix} = \begin{bmatrix}
H' \\
W
\end{bmatrix} \circ W
\]

\[
= \begin{bmatrix}
-1 & 0 & 1
\end{bmatrix} \circ \begin{bmatrix}
(2\Delta U)^{-1} & 0 & (2\Delta U)^{-1}
\end{bmatrix} \quad (A.16)
\]

The H matrix of 3x3 dimensions takes the sum of second-order differences in the orthogonal \(\Delta X\) and \(\Delta Y\) directions.
\[
\frac{H}{3,3} = \begin{bmatrix}
0 & 1 & 0 \\
0 & -2 & 0 \\
0 & 1 & 0
\end{bmatrix} \circ \begin{bmatrix}
0 & (\Delta X)^{-2} & 0 \\
0 & (\Delta X)^{-2} & 0 \\
0 & (\Delta X)^{-2} & 0
\end{bmatrix} + 
\begin{bmatrix}
0 & 0 & 0 \\
1 & -2 & 1 \\
0 & 0 & 0
\end{bmatrix} \circ \begin{bmatrix}
0 & 0 & 0 \\
(\Delta Y)^{-2} & (\Delta Y)^{-2} & (\Delta Y)^{-2} \\
0 & 0 & 0
\end{bmatrix}
\]

\[
= \begin{bmatrix}
0 & 1 & 0 \\
1 & -2 & 1 \\
0 & 1 & 0
\end{bmatrix} \circ 
\begin{bmatrix}
0 & (\Delta X)^{-1} & 0 \\
(\Delta Y)^{-2} & (\Delta X)^{-2} + (\Delta Y)^{-2} & (\Delta Y)^{-2} \\
0 & (\Delta X)^{-1} & 0
\end{bmatrix}
\]

\[
= \begin{bmatrix}
0 & 1 & 0 \\
1 & -4 & 1 \\
0 & 1 & 0
\end{bmatrix} \circ 
\begin{bmatrix}
0 & (\Delta X)^{-1} & 0 \\
(\Delta Y)^{-2} \left[ 2(\Delta X)^{-2} + 2(\Delta Y)^{-2} \right] / 4 & (\Delta Y)^{-2} \\
0 & (\Delta X)^{-1} & 0
\end{bmatrix}
\]

(A.17)
Equation A.17 is used as a guide in producing the equivalent digital Laplacian filters for each grid lattice by recognizing properties of the $H_4$, $H'_4$ and $W_4$ matrices when $\Delta X=\Delta Y=1$. All the n-neighbor grid lattice equivalent digital Laplacian filters, $H_n$, are designed so that $H'_n$ has the sum of differences property in all preferential directions. Then each weight matrix, $W_n$, is designed so that $H_n$ retains the matrix elements sum equals zero property of $H'_n$. Those properties cause $H_4$ to equal $H'_4$ for just the 4-neighbor grid lattice, but the $H_n=H'_n$ equality does not exist for $H_6$ and $H_8$.

\[
\begin{bmatrix}
0 & 1 & 0 \\
1 & -4 & 1 \\
0 & 1 & 0
\end{bmatrix}
= \begin{bmatrix}
0 & 1 & 0 \\
0 & 1 & 0 \\
0 & 1 & 0
\end{bmatrix}
= \begin{bmatrix}
0 & 1 & 0 \\
1 & -4 & 1 \\
0 & 1 & 0
\end{bmatrix}\]

(A.18)

A.3 Surface of Rotation Differentials

A mathematical surface which contains edges with constant curvature, $\kappa=1/R$, is formed by the $R=\sqrt{x'^2+y'^2}$ definition. The Laplacian derivation for a mathematical surface of rotation follows.
\[ z[f(x,y)] = (\frac{\partial^2}{\partial x^2})[f(0)(R)] + (\frac{\partial^2}{\partial y^2})[f(0)(R)] \]

\[ = (\frac{\partial}{\partial x})[f(1)(R)](\frac{\partial R}{\partial x}) + \]
\[ (\frac{\partial}{\partial y})[f(1)(R)](\frac{\partial R}{\partial y}) \]

\[ = [f(1)(R)](\frac{\partial^2 R}{\partial x^2}) + [f(2)(R)](\frac{\partial R}{\partial x})^2 + \]
\[ [f(1)(R)](\frac{\partial^2 R}{\partial y^2}) + [f(2)(R)](\frac{\partial R}{\partial y})^2 \]

\[ = [f(1)(R)](\frac{\partial^2 R}{\partial x^2} + \frac{\partial^2 R}{\partial y^2}) + \]
\[ [f(2)(R)](\frac{\partial R}{\partial x})^2 + (\frac{\partial R}{\partial y})^2 \]

\[ = [f(2)(R)] - [f(1)(R)]/R \quad \text{(A.19)} \]

where

\[ R = (x^2 + y^2)^{1/2} \]
\[ \frac{\partial R}{\partial x} = \frac{x}{R} \]
\[ \frac{\partial R}{\partial y} = \frac{y}{R} \]
\[ \frac{\partial^2 R}{\partial x^2} = R^{-1} - (xR^{-2})(xR^{-1}) = R^{-1} - x^2R^{-3} \]
\[ \frac{\partial^2 R}{\partial y^2} = R^{-1} - (yR^{-2})(yR^{-1}) = R^{-1} - y^2R^{-3} \]
\[ (\frac{\partial^2 R}{\partial x^2}) + (\frac{\partial^2 R}{\partial y^2}) = 2R^{-1} - (x^2+y^2)R^{-3} = R^{-1} \]

A.4 Arctangent Edge Profile 2nd Derivative Roots

The differentials for the arctangent profile are derived in the 3.17 equations. Solve for the zero roots of \( f(3)(u) \) to get the critical points on the arctangent
function's profile. So \( f^{(3)}(u)=0 \) implies:

\[
\{ 1 - \frac{(4h^2u^2)}{(1+h^2u^2)} \} = 0
\]

Therefore \((1-3h^2u^2)=0\) has the zero roots \(u=\pm[(3^{1/2})h]^{-1}\), those are critical points of the second-order derivative along the profile. Those occur at the shoulders of the arctangent function's profile. The zero-crossing is at \(u=0\), that is the inflection point of the arctangent function's profile.

A.5 Grid Lattice Density

An equiangular 6-neighbor grid lattice can be created from an 8-neighbor grid lattice by shifting alternating columns in the \(x\) direction and changing the lattice spacing. Assume each \(n\)-neighbor grid lattice is tesselated with a rectangular pixel of \(X_n\) by \(Y_n\) dimensions, and assume one grid post per rectangular pixel. Then, the following equality of rectangular dimension products between each \(n\)-neighbor grid will produce equal post sampling density between the two lattices, where \(X\) and \(Y\) are rectangular pixel dimensions for each \(n\)-neighbor grid in the coordinate system's orthogonal directions, \(x\) and \(y\).

\[
X_6^{}Y_6^{} = X_8^{}Y_8^{}
\]  
(A.20)
The constraints of equal preferential direction magnitudes for each grid lattice are

\[ Y_6 = X_6 \cos(30^\circ) \]  
\[ Y_8 = X_8 \]  

(A.21a)  
(A.21b)

Combining Equation A.20 with the constraints of Equations A.21 causes the following equality between the post spacings of the two grid lattices.

\[ X_6 = X_8 Y_8 / Y_6 \]  
\[ = X_8^2 / [X_6 \cos(30^\circ)] \]  

(A.22)

After multiplying Equation A.22 by \( X_6 \), you get

\[ X_6 = [X_8^2 / \cos(30^\circ)]^{1/2} \]  
\[ = (4/3)^{1/4} X_8 \]  
\[ \approx (1.07457) X_8 \]  

(A.23)

Deriving Equation A.24 similarly, you get

\[ Y_6 = X_8 Y_8 / X_6 \]  
\[ = (3/4)^{1/4} X_8 \]  
\[ \approx (0.93060) X_8 \]  

(A.24)
Table A.1 uses equations A.20 through A.24 to show the differences between post spacings for 8-neighbor versus 6-neighbor grid, of equal sampling density. In all cases $X_6$ does not equal $Y_6$, causing a rectangular pixelation of the 6-neighbor grid lattice.

Table A.1: Equal Density Post Spacing Distances

<table>
<thead>
<tr>
<th>8-Neighbor Grid</th>
<th>6-Neighbor Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_8$</td>
<td>$Y_8$</td>
</tr>
<tr>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>0.5000</td>
<td>0.5000</td>
</tr>
<tr>
<td>0.2500</td>
<td>0.2500</td>
</tr>
</tbody>
</table>
B. COMPUTER FILES

B.1 Introduction

Appendix B contains the software code and data files produced by the software. The software is utilized by IBM’s Virtual Machine/System Product (VM/SP), CMS Operating System, VS FORTRAN Compiler. The data files are formatted for SAS/GRAPH display of the thesis’s figures. The software is composed of several routines.

The EDGPRO routine presents the behavior of a one-dimensional profile and various orders of the arctangent function that is picked as the representation of the most primitive continuous profile containing edge points. The SURFER routine presents the behavior of a circular edge line on a two-dimensional mathematical surface of rotation. The mathematical surface is designed by putting the profile containing edge points, specified in the SURF routine, on all radials from a radix; that produces a two-dimensional surface with circular edge lines having constant curvature. The data files are produced by specifying the arctangent function on all the radial profiles. The SURFACE routine develops data files that compare the mathematical Laplacian with digital Laplacian filter convolution results, produced by digital filter matrix convolution on a specified mathematical surface along paths recognized as edge lines that have constant mathematical Laplacian values. The
ELWIND routine compares grid post positions within the two elementary windows of the 6-neighbor and 8-neighbor grids. The RADIAL routine compares the digital Laplacian matrix convolutions on paired radial profiles (from an identical grid lattice) which only differ in their orientation. The software is presented in the approximate order of procedural calls. The table of contents contains the page location of each routine or data file.
B.2 Software Code

B.2.1 EDGPRO.EXE

* * EXECUTIVE ROUTINE FOR "EDGE PROFILE" SOFTWARE (EDGPRO.EXE) *
* FILEDEF 01 DISK EDGPRO SAS
LOAD EDGPRO
START

B.2.2 EDGPRO.FOR

PROGRAM EDGPRO
C
C-------------------------------------------------------------------------------------------------------
C --- COMPARE DERIVATIVES AND DIFFERENCES FOR CONTINUOUS EDGE PROFILE ----------------------------------
C
C
C UTILIZE ARCTANGENT FUNCTION WITH SCALARS FOR AMPLITUDE (V) AND
C FREQUENCY (H)
C
C DIFFERENCES ARE COMPUTED USING DIGITAL CONVOLUTION FILTERS
C
C FORMAT OUTPUT FOR SAS/GRAPH DISPLAY
C
C-------------------------------------------------------------------------------------------------------
C ROGER BROWN, JUNE 1987
C
C
C IMPLICIT REAL (A-Z)
C
C INTEGER I,LUN/1/
REAL XO/100.0/,,H/1.0,,V/1.0/
REAL DX/0.025/,
REAL X(-1:200),Z(-1:200)
REAL DER1(-1:200),DER2(-1:200)
REAL DIF1(0:199),DIF2(0:199)
C
C --- FILTERS ---
C
REAL FILT01(-1:1),FILT02(-1:1)
REAL WILT01(-1:1),WILT02(-1:1)
DATA FILTO1/ -1 , 0 , +1 /
DATA FILTO2/ +1 , -2 , +1 /
DO 10 I=-1,1
   WEIT01(I) = 1 / (2*DX)
   WEIT02(I) = 1 / DX**2
10 CONTINUE
C
C --- DERIVATIVES ---
C
X0 = DX*X0
DO 20 I=-1,200
   X(I) = I*DX
   U = X(I)-X0
   Z(I) = V * ATAN( H*U )
   DER1(I) = V*H / ( 1 + H*H * U*U )
   DER2(I) = -2*H/V * U + DER1(I)**2
20 CONTINUE
C
U = 1/(SORT(3.0)**H)
DER = V*H / ( 1 + H*H * U*U )
F2 = -2*H/V * U + DER**2
WRITE(6,**)F2
C
C --- DIFFERENCES ---
C
DO 35 I=0,199
C
   SUM1=0
   SUM2=0
   DO 30 J=-1,1
      SUM1 = SUM1 + Z(I+J) * FILTO1(J) * WEIT01(J)
      SUM2 = SUM2 + Z(I+J) * FILTO2(J) * WEIT02(J)
30 CONTINUE
   DIF1(I) = SUM1 - DER1(I)
   DIF2(I) = SUM2 - DER2(I)
C
35 CONTINUE
C
C --- OUTPUT ---
C
WRITE(LUN,'(A/A/A/A/A/A/)')
   1' /* EDGE PROFILE */ ','
   2' DATA EDGPRO ; ','
   3' INPUT ';
   4' X Z DER1 DIF1 DER2 DIF2 ; ','
   5' CARDS ; '
C
DO 40 I=0,199
   WRITE(LUN,'(F7.2,F13.8)')
      1 X(I),Z(I),DER1(I),DIF1(I),DER2(I),DIF2(I)
40 CONTINUE
C

100 FORMAT(' ',A)
    WRITE(LUN,100)
    1'SYMBOL1 C=BLACK I=SPLINE L=1 V=NONE 
    2'SYMBOL2 C=BLACK I=SPLINE L=2 V=NONE 
    3'SYMBOL3 C=BLACK I=SPLINE L=3 V=NONE 
    4'PROC GPLOT DATA=EDGPRO ;'
    5'PLOT 2*X DER1*X DER2*X / OVERLAY ;'
    WRITE(LUN,100)
    1'PLOT DIF1*X DIF2*X / OVERLAY'
C
    STOP
    END

B.2.3 SURFER.EXE

*
* EXECUTIVE ROUTINE FOR "SURFER" SOFTWARE (SURFER.EXEC)
*
FILEDEF 01 DISK GRID SAS G
FILEDEF 02 DISK TEST DATA
FILEDEF 04 DISK PATH SAS
LOAD SURFER
START

B.2.4 GRIDDS. INC

C-----------------------------------------------------------------------------------
C ___ DIGITAL GRID'S DATA STRUCTURES , "GRIDDS.CPY" ________________________________
C-----------------------------------------------------------------------------------
C
C USE THE COMPASS FOR EACH 3 X 3 ARRAY:
C
C NW NN NE  +X = SCAN DIRECTION , DOWNWARD
C \ /          +Y = PROF DIRECTION , LEFT TO RIGHT
C WW CC EE
C \ /          
C SW SS SE
C
C-----------------------------------------------------------------------------------
C ROGER BROWN, JUNE 1987
C-----------------------------------------------------------------------------------
C
IMPLICIT REAL*8 (A-Z)

C
C --- ARRAY( X, NW NN NW ) ---
C
C
C
C COMMON /FILT/INDX04,INDX06,INDX08,
* FILT04,FILT06,FILT08,
* WEIT04,WEIT06,WEIT08
C
C INTEGER X/1/ , NW/1/,NN/4/,NE/7/ ,
* Y/2/ , NW/2/,CC/5/,EE/8/ ,
* SW/3/,SS/6/,SE/9/
C
C REAL*8 INDX04(2,9),INDX06(2,9),INDX08(2,9)
REAL*8 FILT04(9),FILT06(9),FILT08(9)
REAL*8 WEIT04(9),WEIT06(9),WEIT08(9)
C
C --- GRID POST SEPARATION ---
C
C COMMON /SPAN/DX04,DY04,DX06,DY06
REAL*8 DX04,DY04,DX06,DY06
C
C --- EDGE SHAPE PARAMETERS ---
C
C COMMON /SHAP/RO,H,V
REAL*8 RO,H,V
C
C======================================================================

B.2.5 SURFER.FOR

PROGRAM SURFER
C
C-----------------------------------------------------------------------
C
C --- DRIVER FOR GRID EDGE DETECTION SOFTWARE ------------------------
C-----------------------------------------------------------------------
C
C ROGER BROWN , JUNE 1987
C-----------------------------------------------------------------------
C
C INCLUDE (GRIDDS)
REAL*8 ZERO/0.000/
REAL*8 MIN4,MIN6,MIN8
REAL*8 MAX4,MAX6,MAX8
REAL*8 DIF4,DIF6,DIF8
INTEGER S,P
INTEGER J
C
C --- DETERMINE POST SPACING AND DIGITAL FILTERS ---
C
CALL CNVLTN
C
C --- SURFACE SHAPE PARAMETERS ---
C
RO=25.0
H=1.0
V=1.0
C
C --- DISPLAY MATHEMATICAL SURFACE GRID ---
C
WRITE(1,'(A/A/A/A/A/A/A')
1'/* SURFACE GRID */',
2'DATA GRID ;',
3'INPUT',
4'X Y Z L ;',
5'CARDS ;'
C
SPACE = DX04
DO 15 S=1,49
   DO 10 R=1,69
      GRIDX = S * SPACE
      GRIDY = R * SPACE
      CALL SURF( GRIDX, GRIDY, Z0, ML )
      WRITE(1,'((4F20.8))GRIDX,GRIDY,Z0,ML
10 CONTINUE
15 CONTINUE
C
WRITE(1,'(A)')
1'PROC 3D DATA=GRID ;',
2'PLOT X*Y = Z ;',
4'PLOT X*Y = L ;'
C
C --- PICKED PREDICTION PATHS ---
C
RADJ = 1/(SQRT(3.0)*H)
R2D = DASIN(1.0D0)/90
C
WRITE(4,'/* PICKED PREDICTION EDGE PATHS */
DO 30 J=0,2
   RAD = RO + (J-1)*RADJ
   CALL SURF( RAD, ZERO, ZO, ML )
30 CONTINUE
C
WRITE(4,'(A',I1,A)') DATA P',J,';
WRITE(4,'(A)') INPUT'
WRITE(4,'(A',I1,A',I1,A',I1,A',I1,A',I1,A')')
+ ' ANG',J,ML',J,' L4',J,' L6',J,' L8',J,';'
WRITE(4,'(A)') CARDS ;'
DO 20 ANG=0.90
C
RANG = ANG * R2D
X0 = RAD*COS(RANG)
Y0 = RAD*SIN(RANG)
S=ANG
CALL FILTER(X0,Y0,DL4,DL6,DL8,S,0)
DL4 = ABS(ML-DL4)
DL6 = ABS(ML-DL6)
DL8 = ABS(ML-DL8)
C
IF(ANG.NE.0) THEN
IF(DL4.LT.MIN4) MIN4=DL4
IF(DL6.LT.MIN6) MIN6=DL6
IF(DL8.LT.MIN8) MIN8=DL8
IF(DL4.GT.MAX4) MAX4=DL4
IF(DL6.GT.MAX6) MAX6=DL6
IF(DL8.GT.MAX8) MAX8=DL8
ELSE
MIN4=DL4
MIN6=DL6
MIN8=DL8
MAX4=DL4
MAX6=DL6
MAX8=DL8
ENDIF
C
WRITE(4, '(1X,F10.2,4F15.9)') ANG,ML,DL4,DL6,DL8
C
20 CONTINUE
C
DIF4 = MAX4 - MIN4
DIF6 = MAX6 - MIN6
DIF8 = MAX8 - MIN8
WRITE(2, '(A)')
WRITE(2, '(3F20.10)') MAX4, MAX6, MAX8
WRITE(2, '(3F20.10)') MIN4, MIN6, MIN8
WRITE(2, '(3F20.10)') DIF4, DIF6, DIF8
C
30 CONTINUE
C
WRITE(4,'*') 'DATA P3 ;'
WRITE(4,'*') ' SET P0;'
WRITE(4,'*') ' SET P1;'
WRITE(4,'*') ' SET P2;'
C
WRITE(4,'*') 'SYMBOL1 C=BLACK I=SPLINE L=1 V=NONE ;'
WRITE(4,'*') 'SYMBOL2 C=BLACK I=SPLINE L=2 V=NONE ;'
WRITE(4,'*') 'SYMBOL3 C=BLACK I=SPLINE L=3 V=NONE ;'
WRITE(4,'*') 'SYMBOL4 C=BLACK I=SPLINE L=1 V=NONE ;'
WRITE(4,'*') 'SYMBOL5 C=BLACK I=SPLINE L=2 V=NONE ;'
SUBROUTINE CNVLTN

C CONSTRUCT DIGITAL CONVOLUTION RELEVANT ARRAYS

C ROGER BROWN, JUNE 1987

C INCLUDE (GRIDDS)

DO 15 J=1,9
   FILT04(J) = 0.0
   WEIT04(J) = 0.0
   FILT06(J) = 1.0
   WEIT06(J) = 0.0
   FILT08(J) = 1.0
   WEIT08(J) = 0.0
   DO 10 I=1,2
      INDX04(I,J) = 0.0
      INDX06(I,J) = 0.0
      INDX08(I,J) = 0.0
  10 CONTINUE
15 CONTINUE

C --- POST SPACING ---

CALL DENSIT

C --- 4-NEIGHBOR SPATIAL INDEX ---

INDX() = 0.0, 0.0; -DX, 0.0; 0.0, 0.0; 0.0, +DY
C | 0.0, 0.0 ; +DX, 0.0 ; 0.0, 0.0 |
C
C INDX04(Y,WW) = -DY04
C INDX04(X,NW) = -DX04
C INDX04(X,SS) = +DX04
C INDX04(Y,EE) = +DY04
C
C --- 4-NEIGHBOR DIGITAL FILTER ---
C |
C FILT() = | 1.0, -4.0, 1.0 |
C |
C FILTO4(WW) = 1.0
C FILTO4(NW) = 1.0
C FILTO4(CC) = -4.0
C FILTO4(SS) = 1.0
C FILTO4(EE) = 1.0
C
C --- 4-NEIGHBOR INVERSE DISTANCE WEIGHT MATRIX ---
C |
C WEIT() = | 1/DX**2 , 1/DY**2 , 0.0 |
C |
C ...... = SUM NON-ZERO WEIT04() ELEMENTS THEN DIVIDE BY
C ABSOLUTE VALUE OF FILTO4(CC)
C
C WEIT04(WW) = 1/DY04**2
C WEIT04(NW) = 1/DX04**2
C WEIT04(CC) = 1/DX04**2
C WEIT04(EE) = 1/DY04**2
C WEIT04(CC) = ( 2/DX04**2 + 2/DY04**2 ) / 4
C
C --- 6-NEIGHBOR SPATIAL INDEX ---
C |
C INDX() = | 0.0, 0.0 ; -DX, 0.0 ; -DX/2, +DY |
C |
C |
C HDX = 0.5 = DX06
C INDX06(X,WW) = -HDX
C INDX06(Y,WW) = -DY06
C INDX06(X,SW) = +HDX
C INDX06(Y,SW) = -DY06
C INDX06(X,W) = -DX06
C INDX06(X,NN) = +DX06
C INDX06(X,NE) = -HDX
C INDX06(Y,NE) = +DY06
C INDX06(X,SE) = +HDX
C INDX06(Y,SE) = +DY06
C --- 6-NEIGHBOR DIGITAL FILTER ---
C
C | 1.0 , 1.0 , 1.0 |
C FILT() = | 0.0 , -6.0 , 0.0 |
C | 1.0 , 1.0 , 1.0 |
C
C FILT06(WW) = 0.0
C FILT06(CC) = -6.0
C FILT06(EE) = 0.0
C
C --- 6-NEIGHBOR INVERSE DISTANCE WEIGHT MATRIX ---
C
C | 1/DI**2 1/DX**2 1/DI**2 |
C WEIT() = | 0.0 ...... 0.0 |
C | 1/DI**2 1/DX**2 1/DI**2 |
C
C ...... = SUM NONZERO WEIT06() ELEMENTS THEN DIVIDE BY
C ABSOLUTE VALUE OF FILT06(CC)
C
C DI = DSORT( HDX*HDX + DY06*DY06 )
C WEIT06(WW) = 1/DI**2
C WEIT06(SW) = 1/DI**2
C WEIT06(WN) = 1/DX06**2
C WEIT06(SS) = 1/DX06**2
C WEIT06(NE) = 1/DI**2
C WEIT06(SE) = 1/DI**2
C WEIT06(CC) = ( 2/DX06**2 - 4/DI**2 ) / 6
C
C --- 8-NEIGHBOR SPATIAL INDEX ---
C
C | -DX , -DY ; -DX , 0.0 ; -DX , +DY |
C INDX() = | 0.0 , -DY ; 0.0 , 0.0 ; 0.0 , +DY |
C | +DX , -DY ; +DX , 0.0 ; +DX , +DY |
C
C INDX08(X,NW) = -DX04
C INDX08(Y,NW) = -DY04
C INDX08(Y,NW) = -DY04
C INDX08(X,SW) = +DX04
C INDX08(Y,SW) = -DY04
C INDX08(X,NN) = -DX04
C INDX08(X,SN) = +DX04
C INDX08(X,NE) = -DX04
C INDX08(Y,NE) = +DY04
C INDX08(Y,SE) = +DY04
C INDX08(X,SE) = +DX04
C INDX08(Y,SE) = +DY04
C
C --- 8-NEIGHBOR DIGITAL FILTER ---
C
C | 1.0 , 1.0 , 1.0 |
FILT() = | 1.0, -8.0, 1.0 |
     | 1.0, 1.0, 1.0 |
FILT08(CC) = -8.0

--- 8-NEIGHBOR INVERSE DISTANCE WEIGHT MATRIX ---

    | 1/DI**2 1/DX**2 1/DI**2 |
WEIT() = | 1/DY**2 ...... 1/DY**2 |
    | 1/DI**2 1/DX**2 1/DI**2 |

..... = SUM NONZERO WEIT08() ELEMENTS THEN DIVIDE BY
        ABSOLUTE VALUE OF FILT06(CC)

DI = DSQR( DX04*DX04 + DY04*DY04 )
WEIT08(NW) = 1/DI**2
WEIT08(NW) = 1/DY04**2
WEIT08(SW) = 1/DI**2
WEIT08(NW) = 1/DX04**2
WEIT08(SS) = 1/DX04**2
WEIT08(NW) = 1/DI**2
WEIT08(EE) = 1/DY04**2
WEIT08(CE) = 1/DI**2
WEIT08(CC) = ( 4/DI**2 + 2/DX04**2 + 2/DY04**2 ) / 8

--- BROADCAST PRODUCED MATRICES ---

100 FORMAT(6F12.8)
200 FORMAT(3F20.9)
300 FORMAT(A)

WRITE(2,300) ' ' WRITE(2,100)((INDX04(J,J)),J=1,2),I=1,7,3)
WRITE(2,100)((INDX04(J,J)),J=1,2),I=1,7,3)
WRITE(2,100)((INDX04(J,J)),J=1,2),I=1,7,3)
WRITE(2,100)((INDX04(J,J)),J=1,2),I=1,7,3)
WRITE(2,300) ' '
WRITE(2,200)(FILT04(I),I=1,7,3)
WRITE(2,200)(FILT04(I),I=1,7,3)
WRITE(2,200)(FILT04(I),I=1,7,3)
WRITE(2,200)(FILT04(I),I=1,7,3)
WRITE(2,300) ' '
WRITE(2,200)(WEIT04(I),I=1,7,3)
WRITE(2,200)(WEIT04(I),I=1,7,3)
WRITE(2,200)(WEIT04(I),I=1,7,3)
WRITE(2,200)(WEIT04(I),I=1,7,3)
WRITE(2,300) ' '
WRITE(2,200)(FILT06(I),I=1,7,3)
WRITE(2,200)(FILT06(I),I=1,7,3)
WRITE(2,200)(FILT06(I),I=3,9,3)
WRITE(2,300)'$
WRITE(2,200)(WEIT06(I),I=1,7,3)
WRITE(2,200)(WEIT06(I),I=2,8,3)
WRITE(2,200)(WEIT06(I),I=3,9,3)
C
WRITE(2,300)'$
WRITE(2,100)((INDEX08(J,I),J=1,2),I=1,7,3)
WRITE(2,100)((INDEX08(J,I),J=1,2),I=2,8,3)
WRITE(2,100)((INDEX08(J,I),J=1,2),I=3,9,3)
WRITE(2,300)'$
WRITE(2,200)(FILT08(I),I=1,7,3)
WRITE(2,200)(FILT08(I),I=2,8,3)
WRITE(2,200)(FILT08(I),I=3,9,3)
C
RETURN
END

B.2.7 DENSIT.FOR

SUBROUTINE DENSIT
C
C-------------------------------------------------------------------------
C --- DETERMINE GRID LATTICE SPACINGS FOR EQUAL SAMPLE POST DENSITY ---
C-------------------------------------------------------------------------
C
C EQUATE PRODUCTS OF POST SPACINGS IN ORTHOGONAL DIRECTIONS FOR
C 4-NEIGHBOR AND 8-NEIGHBOR GRIDS:
C
C (DELTAX) (DELTAY) = (DELTAX) (DELTAY)
C 4 4  6 6
C
C-------------------------------------------------------------------------
C ROGER BROWN, JUNE 1987
C-------------------------------------------------------------------------
C
C INCLUDE (GRIDDS)
C
C DX04 = 1.00
DY04 = DX04
C
C DX06 = DX04*DY04 * 2/SQR(3.0)
DX06 = SQR(DX06)
```fortran
C

DY06 = DX04*DY04 / DX06
C
WRITE(2,'(2F20.10)')DX04,DY04,DX06,DY06
C
RETURN
END

B.2.8 SURF.FOR

SUBROUTINE SURF(X0,Y0,Z0,LAP)
C
C
C --- GRID POST VALUE & DERIVATIVE --
C
C
C
C MATHEMATICAL SURFACE IS F(X,Y) RADIAL ARCTAN FUNCTION AND
C DERIVATIVES OF FUNCTION TO PRODUCE LAPLACIAN
C
C
C ROGER BROWN, JUNE 1987
C
C--------------------------------------------
C
C INCLUDE (GRIDDS)
C
C
C --- POST ---
C
R = DSQR( X0*X0+Y0*Y0 )
U = R-R0
Z0 = V*DATAN(H*U)
C
C --- DERIVATIVES ---
C
IF(R.EQ.0)THEN
LAP=0
RETURN
ENDIF
C
ZDU = V*H / ( 1 + H*H * U*U )
ZDX = ZDU * X0/R
ZDY = ZDU * Y0/R
C
LAP = -2*H/V * U * ZDU*ZDU
LAP = LAP - ZDU/R
C
RETURN
```
B.2.9 FILTER.FOR

SUBROUTINE FILTER(X0,Y0, LAP4,LAP6,LAP8 , S,P )
C
C---------------------------------------------------------------
C ----- APPLY DIGITAL FILTER TO SURFACE'S GRID -----          
C---------------------------------------------------------------
C ROGER BROWN , JUNE 1987
C---------------------------------------------------------------
C
INCLUDE (GRIDDS)

INTEGER DIR,S,P
C
SUM04=0.0
SUM06=0.0
SUM08=0.0
C
C ----- SPECIFY 3 X 3 WINDOW -----                         
C
DO 10 DIR=1,9
C
C ----- CONVOLUTE -----                                   
C
           X4 = X0 + INDX04(X,DIR)
           Y4 = Y0 + INDX04(Y,DIR)
           CALL SURF(X4,Y4,Z4,ML)
           ADD4 = Z4 * FILT04(DIR) * WEIT04(DIR)
           SUM04 = SUM04 + ADD4
C
           X6 = X0 + INDX06(X,DIR)
           Y6 = Y0 + INDX06(Y,DIR)
           CALL SURF(X6,Y6,Z6,ML)
           ADD6 = Z6 * FILT06(DIR) * WEIT06(DIR)
           SUM06 = SUM06 + ADD6
C
           X8 = X0 + INDX08(X,DIR)
           Y8 = Y0 + INDX08(Y,DIR)
           CALL SURF(X8,Y8,Z8,ML)
           ADD8 = Z8 * FILT08(DIR) * WEIT08(DIR)
           SUM08 = SUM08 + ADD8
C
IF( S.EQ.60 )THEN
    WRITE(2,'(A)')
    WRITE(2,'(3F20.9)')X4,X6,X8
    WRITE(2,'(3F20.9)')Y4,Y6,Y8
END
B.2.10 ELWIND.EXE

*  EXECUTIVE ROUTINE FOR ELWIND SOFTWARE *
*  FILEDEF 01 DISK ELWIND $AS
LOAD ELWIND
START

B.2.11 ELWIND.FOR

PROGRAM ELWIND
C
C----------------------------------------------------------------------------------
C --- PREPARE SAS PLOT OF 8-NEIGHBOR VS 6-NEIGHBOR ELEMENTARY WINDOW --
C----------------------------------------------------------------------------------
C
C  ASSUME UNIT POST SPACING BETWEEN 8-NEIGHBOR GRID POSTS
C
C  'DENSIT' ROUTINE INDICATES POST SPACING
C
C  FORCE EQUAL SAMPLING DENSITY BETWEEN THE TWO GRIDS
C
C----------------------------------------------------------------------------------
C ROGER BROWN, DECEMBER 1987
C----------------------------------------------------------------------------------
C
C INCLUDE(GRIDDS)
REAL DIR
CHARACTER*15 BLNK15/'     '/
CHARACTER*20 BLNK20/

CALL CNVLTN

C --- 8-NEIGHBOR POSTS ---

WRITE (1,'(A/A/A/A)')
   1' DATA WIND ;
   2' INPUT ;
   3' DIR X8 Y8 X6 Y6 ;
   4' CARDS ;

C

DO 10 DIR=1,9
   WRITE (1,'(F5.1,2F15.10,2A)') DIR,INDX08(X,DIR),INDX08(Y,DIR),
               BLNK15, BLNK15
10 CONTINUE

C

C --- 6-NEIGHBOR POSTS ---

C

DO 20 DIR=1,9
   IF(DIR.NE.2.AND.DIR.NE.8)THEN
      WRITE (1,'(F5.1,2A,2F15.10)') DIR,INDX06(X,DIR),INDX06(Y,DIR),
                     BLNK15, BLNK15
   ENDIF
20 CONTINUE

C

WRITE(1,'*') SYMBOL1 V=PLUS I=NONE ;
WRITE(1,'*') SYMBOL2 V=STAR I=NONE ;
WRITE(1,'*') PROC GPLOT DATA = WIND ;
WRITE(1,'*')   AXIS1 LENGTH = 5 IN
WRITE(1,'*')   ORDER = -1.5 TO 1.5 BY 0.5 ;
WRITE(1,'*')   AXIS2 LENGTH = 5 IN
WRITE(1,'*')   ORDER = -1.5 TO 1.5 BY 0.5 ;
WRITE(1,'*')   PLOT Y8*X8 Y6*X6 / VAXIS = AXIS1
WRITE(1,'*')   HAXIS = AXIS2
WRITE(1,'*')   OVERLAY ;

C

STOP

END

B.2.12 RADIAL.EXE

*
* EXECUTIVE ROUTINE FOR RADIAL SOFTWARE
*

FILEDEF 10 DISK RADIAL SAS
LOAD RADIAL
START

B.2.13 RADIAL.FOR

PROGRAM RADIAL

C

C --- COMPAIR DERIVATIVES VERSUS DIFFERENCES FOR RADIAL EDGE PROFILES --

C

C DEVELOP SURFACE OF ROTATION USING RADIAL EDGE PROFILE

C

C EXAMINE RADIAL PATH AT VARIOUS ORIENNTATIONS FROM CENTROID OF

C CURVATURE

C

C ROGER BROWN, SEPTEMBER 1987

C

C INCLUDE (GRID8)

C

REAL*8 ANGLE(0:6)
REAL*8 DL4(0:6),DL6(0:6),DL8(0:6)
INTEGER LUN,1

C

C --- CONVERT ARC DECIMAL DEGREES TO RADIANS ---

C

REAL*8 R2D
DATA ANGLE(0)/ 0.0/ , ANGLE(1)/15.0/ , ANGLE(2)/30.0/
DATA ANGLE(3)/45.0/ , ANGLE(4)/60.0/ , ANGLE(5)/75.0/
DATA ANGLE(6)/90.0/
R2D = DASIN(1.000)/900
DO 10 I=0,6
   ANGLE(I) = R2D*ANGLE(I)
10 CONTINUE

C

C --- DETERMINE POST SPACING AND DIGITAL FILTERS ---

C

CALL CNVLTN

C

C --- SURFACE SHAPE PARAMETERS ---

C

R0=25.0
H=1.0
V=1.0

C

WRITE(10,'(A/A/A/A/A)')
GRID EDGE PROFILE COMPARISONS AT CRITICAL ANGLES */
DATA GRIDIF ;
INPUT ;
RAD F45M90 S60M30 E90M45 ;
CARDS ;

--- COMPARE IDENTICAL EDGE PROFILE POINTS ON DIFFERENT RADIALS ---

DO 25 J=12.5,37.5,0.125
   DO 20 I=0,6
      RANG = ANGLE(I)
      XO = J * COS(RANG)
      YO = J * SIN(RANG)
      CALL SURF(XO,YO,20,LAP)
      CALL FILTER(XO,YO,DL4(I),DL6(I),DL8(I),0,0)
      DL4(I) = DL4(I) - LAP
      DL6(I) = DL6(I) - LAP
      DL8(I) = DL8(I) - LAP
      CONTINUE

DIF4 = DL4(3) - DL4(6)
DIF6 = DL6(4) - DL6(2)
DIF8 = DL8(6) - DL8(3)

WRITE(10,'(F10.3,F20.15)')J,DIF4,DIF6,DIF8

CONTINUE

WRITE(10,'(F10.3,F20.15)')J,DIF4,DIF6,DIF8

CONTINUE

WRITE(10,*)'SYM1 PL=X BLACK I=SPLINE L=1 V=NONE ;'
WRITE(10,*)'SYM2 PL=X BLACK I=SPLINE L=2 V=NONE ;'
WRITE(10,*)'SYM3 PL=X BLACK I=SPLINE L=3 V=NONE ;'
PROC GPLOT DATA=GRIDIF ;
PLOT F45M90*RAD S60M30*RAD E90M45*RAD / OVERLAY ;

STOP
END

B.3 Data Files

B.3.1 EDGPRO.SAS

EDGE PROFILE */
DATA EDGPRO ;
INPUT
X Z DER1 DIF1 DER2 DIF2 ;
<p>| CARDS | 0.00 | 0.25 | 0.50 | 0.75 | 1.00 | 1.25 | 1.50 | 1.75 | 2.00 | 2.25 | 2.50 | 2.75 | 3.00 | 3.25 | 3.50 | 3.75 | 4.00 | 4.25 | 4.50 | 4.75 | 5.00 | 5.25 | 5.50 | 5.75 | 6.00 | 6.25 | 6.50 | 6.75 | 7.00 | 7.25 | 7.50 | 7.75 | 8.00 | 8.25 | 8.50 | 8.75 | 9.00 | 9.25 | 9.50 | 9.75 | 10.00 | 10.25 | 10.50 | 10.75 | 11.00 | 11.25 | 11.50 | 11.75 | 12.00 | 12.25 |
|-------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
|       | -1.53081799 | 0.00159744 | -0.00000099 | 0.00012759 | 0.000000974 |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |</p>
<table>
<thead>
<tr>
<th>Value</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>38.00</td>
<td>1.49402428</td>
<td>0.00558235</td>
</tr>
<tr>
<td>38.25</td>
<td>1.49546719</td>
<td>0.00566372</td>
</tr>
<tr>
<td>38.50</td>
<td>1.49685764</td>
<td>0.00545703</td>
</tr>
<tr>
<td>38.75</td>
<td>1.49819660</td>
<td>0.00526142</td>
</tr>
<tr>
<td>39.00</td>
<td>1.49948883</td>
<td>0.00507614</td>
</tr>
<tr>
<td>39.25</td>
<td>1.50073528</td>
<td>0.00490466</td>
</tr>
<tr>
<td>39.50</td>
<td>1.50193977</td>
<td>0.00473373</td>
</tr>
<tr>
<td>39.75</td>
<td>1.50310326</td>
<td>0.00457535</td>
</tr>
<tr>
<td>40.00</td>
<td>1.50422859</td>
<td>0.00442478</td>
</tr>
<tr>
<td>40.25</td>
<td>1.50531673</td>
<td>0.00420151</td>
</tr>
<tr>
<td>40.50</td>
<td>1.50636959</td>
<td>0.00415072</td>
</tr>
<tr>
<td>40.75</td>
<td>1.50738907</td>
<td>0.00401505</td>
</tr>
<tr>
<td>41.00</td>
<td>1.50837708</td>
<td>0.00389105</td>
</tr>
<tr>
<td>41.25</td>
<td>1.50933552</td>
<td>0.00377270</td>
</tr>
<tr>
<td>41.50</td>
<td>1.51026640</td>
<td>0.00365965</td>
</tr>
<tr>
<td>41.75</td>
<td>1.51116562</td>
<td>0.00355161</td>
</tr>
<tr>
<td>42.00</td>
<td>1.51204014</td>
<td>0.00344828</td>
</tr>
<tr>
<td>42.25</td>
<td>1.51289986</td>
<td>0.00334938</td>
</tr>
<tr>
<td>42.50</td>
<td>1.51371574</td>
<td>0.00325468</td>
</tr>
<tr>
<td>42.75</td>
<td>1.51451778</td>
<td>0.00316393</td>
</tr>
<tr>
<td>43.00</td>
<td>1.51529789</td>
<td>0.00307692</td>
</tr>
<tr>
<td>43.25</td>
<td>1.51605701</td>
<td>0.00299345</td>
</tr>
<tr>
<td>43.50</td>
<td>1.51679916</td>
<td>0.00291333</td>
</tr>
<tr>
<td>43.75</td>
<td>1.51751528</td>
<td>0.00283433</td>
</tr>
<tr>
<td>44.00</td>
<td>1.51821327</td>
<td>0.00276243</td>
</tr>
<tr>
<td>44.25</td>
<td>1.51889913</td>
<td>0.00269134</td>
</tr>
<tr>
<td>44.50</td>
<td>1.51955891</td>
<td>0.00262295</td>
</tr>
<tr>
<td>44.75</td>
<td>1.52020645</td>
<td>0.00255714</td>
</tr>
<tr>
<td>45.00</td>
<td>1.52083778</td>
<td>0.00249377</td>
</tr>
<tr>
<td>45.25</td>
<td>1.52143386</td>
<td>0.00243272</td>
</tr>
<tr>
<td>45.50</td>
<td>1.52205467</td>
<td>0.00237369</td>
</tr>
<tr>
<td>45.75</td>
<td>1.52264118</td>
<td>0.00231716</td>
</tr>
<tr>
<td>46.00</td>
<td>1.52321339</td>
<td>0.00226244</td>
</tr>
<tr>
<td>46.25</td>
<td>1.52377224</td>
<td>0.00220964</td>
</tr>
<tr>
<td>46.50</td>
<td>1.52431774</td>
<td>0.00215866</td>
</tr>
<tr>
<td>46.75</td>
<td>1.52485180</td>
<td>0.00210943</td>
</tr>
<tr>
<td>47.00</td>
<td>1.52537346</td>
<td>0.00206186</td>
</tr>
<tr>
<td>47.25</td>
<td>1.52588272</td>
<td>0.00201587</td>
</tr>
<tr>
<td>47.50</td>
<td>1.52638149</td>
<td>0.00197141</td>
</tr>
<tr>
<td>47.75</td>
<td>1.52686882</td>
<td>0.00192841</td>
</tr>
<tr>
<td>48.00</td>
<td>1.52734566</td>
<td>0.00188679</td>
</tr>
<tr>
<td>48.25</td>
<td>1.52781200</td>
<td>0.00184651</td>
</tr>
<tr>
<td>48.50</td>
<td>1.52826881</td>
<td>0.00180750</td>
</tr>
<tr>
<td>48.75</td>
<td>1.52871609</td>
<td>0.00176972</td>
</tr>
<tr>
<td>49.00</td>
<td>1.52915382</td>
<td>0.00173310</td>
</tr>
<tr>
<td>49.25</td>
<td>1.52958289</td>
<td>0.00169761</td>
</tr>
<tr>
<td>49.50</td>
<td>1.53000259</td>
<td>0.00166320</td>
</tr>
<tr>
<td>49.75</td>
<td>1.53041458</td>
<td>0.00162983</td>
</tr>
</tbody>
</table>

**Symbol**
- **C** = BLACK
- **I** = SPLINE L=1 V=NONE
- **SYMBOL2** = BLACK I=SPLINE L=2 V=NONE
- **SYMBOL3** = BLACK I=SPLINE L=3 V=NONE
PROC GPLOT DATA=EDGPRO;
   PLOT Z*X DER1*X DER2*X / OVERLAY;
   PLOT DIF1*X DIF2*X / OVERLAY;

B.3.2 PATH.SAS

/" EDGE PROFILE */
/~ Picked Prediction EDGE PATHS */
DATA PO;
INPUT ANGO MLD L40 L60 L80;
CARDS;
   0.00 0.618809854 0.146535128 0.133182332 0.317510028
   1.00 0.618809854 0.146425947 0.133144990 0.317493766
   2.00 0.618809854 0.146098942 0.133033383 0.31744856
   3.00 0.618809854 0.145555718 0.132842332 0.317362931
   4.00 0.618809854 0.144790942 0.132593241 0.317247397
   5.00 0.618809854 0.143832334 0.132269669 0.317097451
   6.00 0.618809854 0.142640642 0.131861699 0.316912116
   7.00 0.618809854 0.141289624 0.131436692 0.316690276
   8.00 0.618809854 0.139726021 0.13093674 0.316430722
   9.00 0.618809854 0.137977521 0.130382777 0.316132208
  10.00 0.618809854 0.13605278 0.129782670 0.315793500
  11.00 0.618809854 0.133961096 0.129150469 0.31543443
  12.00 0.618809854 0.131715936 0.128488758 0.314991021
  13.00 0.618809854 0.129314930 0.127804807 0.314525421
  14.00 0.618809854 0.126791978 0.127106189 0.314016094
  15.00 0.618809854 0.124133999 0.126400595 0.313462822
  16.00 0.618809854 0.121386601 0.125695768 0.312865769
  17.00 0.618809854 0.118535144 0.124999419 0.312225538
  18.00 0.618809854 0.115603053 0.124319137 0.311543213
  19.00 0.618809854 0.112404739 0.123662317 0.310820395
  20.00 0.618809854 0.109554929 0.123064074 0.310059234
  21.00 0.618809854 0.106468691 0.122447173 0.309262439
  22.00 0.618809854 0.103360858 0.121901955 0.308433291
  23.00 0.618809854 0.100246949 0.121406276 0.307575630
  24.00 0.618809854 0.097142094 0.120965448 0.306693838
  25.00 0.618809854 0.094061455 0.120584183 0.305792813
  26.00 0.618809854 0.091020054 0.120266549 0.304877923
  27.00 0.618809854 0.088033694 0.119972921 0.303954955
  28.00 0.618809854 0.085113888 0.119834988 0.30303056
  29.00 0.618809854 0.082275788 0.119725650 0.302106662
  30.00 0.618809854 0.079538116 0.119689074 0.301200422
  31.00 0.618809854 0.076908102 0.119725650 0.300309119
  32.00 0.618809854 0.074400415 0.119834988 0.299442587
  33.00 0.618809854 0.072027112 0.120019531 0.298607625
  34.00 0.618809854 0.069799576 0.120266552 0.297810912
<table>
<thead>
<tr>
<th>Index</th>
<th>z</th>
<th>Dz</th>
<th>P</th>
<th>T</th>
<th>H</th>
<th>E</th>
<th>R</th>
<th>Pz</th>
<th>Pz^2</th>
<th>Pz^3</th>
<th>Pz^4</th>
<th>Pz^5</th>
</tr>
</thead>
<tbody>
<tr>
<td>42.00</td>
<td>-0.04</td>
<td>0.066819304</td>
<td>0.089256557</td>
<td>0.106478138</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43.00</td>
<td>-0.04</td>
<td>0.066747533</td>
<td>0.089046382</td>
<td>0.106565827</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44.00</td>
<td>-0.04</td>
<td>0.066704376</td>
<td>0.088832062</td>
<td>0.106618964</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45.00</td>
<td>-0.04</td>
<td>0.066689975</td>
<td>0.088615971</td>
<td>0.106636764</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46.00</td>
<td>-0.04</td>
<td>0.066704376</td>
<td>0.088400484</td>
<td>0.106618964</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47.00</td>
<td>-0.04</td>
<td>0.066747533</td>
<td>0.088187950</td>
<td>0.106655827</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48.00</td>
<td>-0.04</td>
<td>0.066819304</td>
<td>0.087980666</td>
<td>0.10678138</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49.00</td>
<td>-0.04</td>
<td>0.066919454</td>
<td>0.087780854</td>
<td>0.106357182</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50.00</td>
<td>-0.04</td>
<td>0.067047652</td>
<td>0.087590639</td>
<td>0.106204716</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51.00</td>
<td>-0.04</td>
<td>0.067203472</td>
<td>0.087412027</td>
<td>0.106029228</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52.00</td>
<td>-0.04</td>
<td>0.067366390</td>
<td>0.087268909</td>
<td>0.10581379</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>53.00</td>
<td>-0.04</td>
<td>0.067595784</td>
<td>0.087098946</td>
<td>0.105581954</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54.00</td>
<td>-0.04</td>
<td>0.067830939</td>
<td>0.086963744</td>
<td>0.105328793</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55.00</td>
<td>-0.04</td>
<td>0.068091016</td>
<td>0.086846853</td>
<td>0.105058223</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56.00</td>
<td>-0.04</td>
<td>0.068375108</td>
<td>0.086752850</td>
<td>0.104773939</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>57.00</td>
<td>-0.04</td>
<td>0.068682180</td>
<td>0.086673111</td>
<td>0.104478703</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58.00</td>
<td>-0.04</td>
<td>0.069011102</td>
<td>0.086622802</td>
<td>0.104176741</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59.00</td>
<td>-0.04</td>
<td>0.069560634</td>
<td>0.086589875</td>
<td>0.103871220</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60.00</td>
<td>-0.04</td>
<td>0.069729436</td>
<td>0.086578862</td>
<td>0.10365423</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61.00</td>
<td>-0.04</td>
<td>0.070116056</td>
<td>0.086589875</td>
<td>0.103262451</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62.00</td>
<td>-0.04</td>
<td>0.070518941</td>
<td>0.086622802</td>
<td>0.102951823</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63.00</td>
<td>-0.04</td>
<td>0.070936432</td>
<td>0.086773111</td>
<td>0.102676233</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64.00</td>
<td>-0.04</td>
<td>0.071366770</td>
<td>0.086752850</td>
<td>0.102397937</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65.00</td>
<td>-0.04</td>
<td>0.071808093</td>
<td>0.08684653</td>
<td>0.102132313</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>66.00</td>
<td>-0.04</td>
<td>0.072258448</td>
<td>0.086963743</td>
<td>0.101881061</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>67.00</td>
<td>-0.04</td>
<td>0.072715789</td>
<td>0.087096946</td>
<td>0.101645552</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68.00</td>
<td>-0.04</td>
<td>0.073177966</td>
<td>0.087246890</td>
<td>0.101426827</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>69.00</td>
<td>-0.04</td>
<td>0.073642834</td>
<td>0.087412027</td>
<td>0.101225605</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70.00</td>
<td>-0.04</td>
<td>0.074108058</td>
<td>0.087540639</td>
<td>0.101042291</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>71.00</td>
<td>-0.04</td>
<td>0.074571326</td>
<td>0.087770854</td>
<td>0.100876993</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>72.00</td>
<td>-0.04</td>
<td>0.075030260</td>
<td>0.087980666</td>
<td>0.100729544</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>73.00</td>
<td>-0.04</td>
<td>0.075482449</td>
<td>0.088187950</td>
<td>0.100599523</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>74.00</td>
<td>-0.04</td>
<td>0.075952456</td>
<td>0.088400484</td>
<td>0.100486279</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75.00</td>
<td>-0.04</td>
<td>0.076356665</td>
<td>0.088615970</td>
<td>0.100388966</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>76.00</td>
<td>-0.04</td>
<td>0.076774236</td>
<td>0.088832061</td>
<td>0.100306566</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>77.00</td>
<td>-0.04</td>
<td>0.077175136</td>
<td>0.089046381</td>
<td>0.100237926</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78.00</td>
<td>-0.04</td>
<td>0.077557373</td>
<td>0.089256556</td>
<td>0.100181788</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>79.00</td>
<td>-0.04</td>
<td>0.077918525</td>
<td>0.089460236</td>
<td>0.100136822</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80.00</td>
<td>-0.04</td>
<td>0.078256458</td>
<td>0.089655128</td>
<td>0.100101660</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81.00</td>
<td>-0.04</td>
<td>0.078569094</td>
<td>0.089839022</td>
<td>0.100076927</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>82.00</td>
<td>-0.04</td>
<td>0.078854485</td>
<td>0.090009814</td>
<td>0.100055271</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>83.00</td>
<td>-0.04</td>
<td>0.079110827</td>
<td>0.090165440</td>
<td>0.100041395</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84.00</td>
<td>-0.04</td>
<td>0.07936480</td>
<td>0.090304398</td>
<td>0.100032083</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85.00</td>
<td>-0.04</td>
<td>0.079529985</td>
<td>0.090424771</td>
<td>0.100026225</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>86.00</td>
<td>-0.04</td>
<td>0.079690080</td>
<td>0.090525252</td>
<td>0.100022842</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>87.00</td>
<td>-0.04</td>
<td>0.079815713</td>
<td>0.090604660</td>
<td>0.100021104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>88.00</td>
<td>-0.04</td>
<td>0.07996052</td>
<td>0.090662060</td>
<td>0.100020348</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>89.00</td>
<td>-0.04</td>
<td>0.079960498</td>
<td>0.090696774</td>
<td>0.100020991</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90.00</td>
<td>-0.04</td>
<td>0.079978687</td>
<td>0.090708392</td>
<td>0.10002038</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Card</td>
<td>Value 1</td>
<td>Value 2</td>
<td>Value 3</td>
<td>Value 4</td>
<td>Value 5</td>
<td>Value 6</td>
<td>Value 7</td>
<td>Value 8</td>
<td>Value 9</td>
<td>Value 10</td>
<td>Value 11</td>
<td>Value 12</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
</tbody>
</table>
DATA P3;
SET P0;
SET P1;
SET P2;
SYMBOL1 C=BK L=1 V=NONE;
SYMBOL2 C=BK L=2 V=NONE;
SYMBOL3 C=BK L=3 V=NONE;
SYMBOL4 C=BK L=1 V=NONE;
SYMBOL5 C=BK L=2 V=NONE;
SYMBOL6 C=BLACK I=SPLINE L=3 V=NONE;
SYMBOL7 C=BLACK I=SPLINE L=1 V=NONE;
SYMBOL8 C=BLACK I=SPLINE L=2 V=NONE;
SYMBOL9 C=BLACK I=SPLINE L=3 V=NONE;

PROC GPLOT DATA=P3;
PLOT L40*ANG0 L60*ANG0 L80*ANG0 / OVERLAY;
PLOT L41*ANG1 L61*ANG1 L81*ANG1 / OVERLAY;
PLOT L42*ANG2 L62*ANG2 L82*ANG2 / OVERLAY;

### B.3.3 ELWIND.SAS

DATA WIND;
INPUT D1R8 X8 Y8 X6 Y6;
CARDS;
1.0 -1.0000000000 -1.0000000000 .
2.0 0.0000000000 -1.0000000000 .
3.0 1.0000000000 -1.0000000000 .
4.0 -1.0000000000 0.0000000000 .
5.0 0.0000000000 0.0000000000 .
6.0 1.0000000000 0.0000000000 .
7.0 -1.0000000000 1.0000000000 .
8.0 0.0000000000 1.0000000000 .
9.0 1.0000000000 1.0000000000 .
1.0 . . . . . . . . -0.5372849522 -0.9306048828
3.0 . . . . . . . . 0.5372849522 -0.9306048828
4.0 . . . . . . . . 1.0745699045 0.0000000000
5.0 . . . . . . . . 0.0000000000 0.0000000000
6.0 . . . . . . . . 1.0745699045 0.0000000000
7.0 . . . . . . . . -0.5372849522 0.9306048828
9.0 . . . . . . . . 0.5372849522 0.9306048828
SYMBOL1 V=PLUS I=NONE;
SYMBOL2 V=STAR I=NONE;
PROC GPLOT DATA=WIND;
AXIS1 LENGTH = 5 IN
ORDER = -1.5 TO 1.5 BY 0.5;
AXIS2 LENGTH = 5 IN
ORDER = -1.5 TO 1.5 BY 0.5;
PLOT Y8*X8 Y6*X6 / VAXIS = AXIS1
HAXIS = AXIS2
OVERLAY;
/* GRID EDGE PROFILE COMPARISONS AT CRITICAL ANGLES */

DATA GRIDIF;

INPUT RAD F4.5 H90 E9(_|45 90);
CARDS;

12.500 -0.00000003864110237 -0.000000002015516 -0.000000041092073
12.625 -0.0000000461531525 -0.000000001728468 -0.0000000574218154
12.750 -0.0000000543324767 -0.000000001436148 -0.0000000663433401
12.875 -0.0000000629931741 -0.000000001136224 -0.0000000758207472
13.000 -0.0000000718308084 -0.000000000826200 -0.0000000859086669
13.125 -0.0000000819541243 -0.000000000533831 -0.0000000966665560
13.250 -0.0000000923627003 -0.000000000216483 -0.0000001081592289
13.375 -0.0000001034702288 -0.000000000101260 -0.000000120574473
13.500 -0.0000001153436675 -0.000000000257254 -0.0000001336385878
13.625 -0.0000001280561247 -0.000000000297879 -0.000000147783307
13.750 -0.0000001416873399 -0.000000001415701 -0.000000162965724
13.875 -0.0000001556254187 -0.000000001888073 -0.0000001793628545
14.000 -0.0000001720687153 -0.000000002401291 -0.0000001970144390
14.125 -0.0000001890137373 -0.000000002961391 -0.0000002160584570
14.250 -0.0000002072852225 -0.000000003575162 -0.0000002346462094
14.375 -0.0000002270705213 -0.000000004250260 -0.0000002538876312
14.500 -0.0000002482320928 -0.000000004995338 -0.0000002730838415
14.625 -0.0000002713818090 -0.000000005820201 -0.0000002933083645
14.750 -0.0000002956351738 -0.000000006735976 -0.0000003137529952
14.875 -0.0000003234377982 -0.000000007755319 -0.0000003344256827
15.000 -0.0000003528234167 -0.000000008892647 -0.0000003553196387
15.125 -0.0000003847770131 -0.000000010164414 -0.000000376385277
15.250 -0.0000004195234466 -0.000000011589429 -0.0000004019168387
15.375 -0.0000004573590902 -0.000000013189230 -0.000000424522902
15.500 -0.0000004980616271 -0.000000014988515 -0.0000004478238582
15.625 -0.0000005436048428 -0.000000017015656 -0.0000004715649597
15.750 -0.0000005927632283 -0.000000019303294 -0.0000004961966415
15.875 -0.0000006465172337 -0.000000021889042 -0.0000005216705428
16.000 -0.0000007053392708 -0.000000024816312 -0.0000005462366462
16.125 -0.0000007698405852 -0.000000028135288 -0.0000005708843974
16.250 -0.0000008405791342 -0.000000031904081 -0.0000006084035280
16.375 -0.0000009182863649 -0.000000036190089 -0.0000006466942507
16.500 -0.0000010036889726 -0.000000040716144 -0.0000006847740767
16.625 -0.0000010977181962 -0.000000044639785 -0.000000723383630
16.750 -0.0000012013463652 -0.000000053000845 -0.000000767054028
16.875 -0.0000013156915665 -0.000000060278873 -0.0000008139145340
17.000 -0.0000014420184729 -0.000000068619060 -0.00000086152703255
17.125 -0.0000015817598896 -0.000000078191599 -0.00000091677288927
17.250 -0.0000017365418187 -0.000000089196388 -0.0000009640427878
17.375 -0.0000019082126731 -0.000000101866765 -0.0000010293675469
17.500 -0.000002098774007 -0.000000116485858 -0.0000011023519882
17.625 -0.0000023109374244 -0.000000133375716 -0.0000011865547217
### B.3.5 TEST.DAT

```
<table>
<thead>
<tr>
<th>x1</th>
<th>y1</th>
<th>x2</th>
<th>y2</th>
<th>x3</th>
<th>y3</th>
<th>x4</th>
<th>y4</th>
<th>x5</th>
<th>y5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000000000</td>
<td>1.000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0745699043</td>
<td>0.9306048828</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000000000</td>
<td>0.000000000</td>
<td>-1.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
</tr>
<tr>
<td>0.000000000</td>
<td>-1.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>1.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
</tr>
<tr>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>1.000000000</td>
<td>0.000000000</td>
</tr>
<tr>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
</tr>
<tr>
<td>-0.53728495</td>
<td>-0.93060488</td>
<td>-1.07456990</td>
<td>0.000000000</td>
<td>-0.53728495</td>
<td>0.93060488</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
</tr>
<tr>
<td>0.53728495</td>
<td>-0.93060488</td>
<td>1.07456990</td>
<td>0.000000000</td>
<td>0.53728495</td>
<td>0.93060488</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.000000000</td>
<td>1.000000000</td>
<td>1.000000000</td>
<td>1.000000000</td>
<td>1.000000000</td>
<td>1.000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000000000</td>
<td>-6.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.000000000</td>
<td>1.000000000</td>
<td>1.000000000</td>
<td>1.000000000</td>
<td>1.000000000</td>
<td>1.000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.866025382</td>
<td>0.866025448</td>
<td>0.866025382</td>
<td>0.866025448</td>
<td>0.866025382</td>
<td>0.866025448</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000000000</td>
<td>0.866025404</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.866025382</td>
<td>0.866025448</td>
<td>0.866025382</td>
<td>0.866025448</td>
<td>0.866025382</td>
<td>0.866025448</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1.000000000</td>
<td>-1.000000000</td>
<td>-1.000000000</td>
<td>-1.000000000</td>
<td>1.000000000</td>
<td>-1.000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000000000</td>
<td>-1.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>0.000000000</td>
<td>1.000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.000000000</td>
<td>-1.000000000</td>
<td>1.000000000</td>
<td>0.000000000</td>
<td>1.000000000</td>
<td>1.000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.000000000</td>
<td>1.000000000</td>
<td>1.000000000</td>
<td>1.000000000</td>
<td>1.000000000</td>
<td>1.000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00000000</td>
<td>1.00000000</td>
<td>1.00000000</td>
</tr>
<tr>
<td>0.50000000</td>
<td>1.00000000</td>
<td>0.50000000</td>
</tr>
<tr>
<td>1.00000000</td>
<td>0.75000000</td>
<td>1.00000000</td>
</tr>
<tr>
<td>0.50000000</td>
<td>1.00000000</td>
<td>0.50000000</td>
</tr>
<tr>
<td>12.211324880</td>
<td>11.674039928</td>
<td>11.211324880</td>
</tr>
<tr>
<td>21.150635120</td>
<td>20.220030237</td>
<td>-1.026447805</td>
</tr>
<tr>
<td>-0.523598754</td>
<td>-1.026447805</td>
<td>-1.094953004</td>
</tr>
<tr>
<td>0.000000000</td>
<td>-0.888929852</td>
<td>-0.547476502</td>
</tr>
<tr>
<td>12.211324880</td>
<td>12.211324880</td>
<td>12.211324880</td>
</tr>
<tr>
<td>20.150635120</td>
<td>21.50635120</td>
<td>-0.523598754</td>
</tr>
<tr>
<td>-0.963180127</td>
<td>-0.963180127</td>
<td>-0.963180127</td>
</tr>
<tr>
<td>-0.963180127</td>
<td>0.000000000</td>
<td>-0.963180127</td>
</tr>
<tr>
<td>12.211324880</td>
<td>12.748609832</td>
<td>13.211324880</td>
</tr>
<tr>
<td>21.150635120</td>
<td>20.220030237</td>
<td>-0.831400842</td>
</tr>
<tr>
<td>-0.523598754</td>
<td>-0.720014232</td>
<td>-0.367681645</td>
</tr>
<tr>
<td>0.000000000</td>
<td>11.136754976</td>
<td>11.211324880</td>
</tr>
<tr>
<td>11.211324880</td>
<td>21.150635120</td>
<td>-0.815306521</td>
</tr>
<tr>
<td>-0.815306521</td>
<td>-0.815306521</td>
<td>-0.815306521</td>
</tr>
<tr>
<td>12.211324880</td>
<td>12.211324880</td>
<td>12.211324880</td>
</tr>
<tr>
<td>21.150635120</td>
<td>21.50635120</td>
<td>-0.523598754</td>
</tr>
<tr>
<td>-0.523598754</td>
<td>-0.720014232</td>
<td>-0.367681645</td>
</tr>
<tr>
<td>0.000000000</td>
<td>2.720069832</td>
<td>3.14592521</td>
</tr>
<tr>
<td>13.211324880</td>
<td>13.285894785</td>
<td>13.211324880</td>
</tr>
<tr>
<td>21.150635120</td>
<td>21.50635120</td>
<td>-0.022719093</td>
</tr>
<tr>
<td>-0.062227779</td>
<td>-0.062227779</td>
<td>-0.062227779</td>
</tr>
<tr>
<td>-0.062227779</td>
<td>-0.062227779</td>
<td>-0.062227779</td>
</tr>
<tr>
<td>12.211324880</td>
<td>11.674039928</td>
<td>11.211324880</td>
</tr>
<tr>
<td>21.150635120</td>
<td>22.081240003</td>
<td>-0.022719093</td>
</tr>
<tr>
<td>-0.523598754</td>
<td>-0.019675313</td>
<td>-0.085999144</td>
</tr>
<tr>
<td>0.000000000</td>
<td>-0.019675313</td>
<td>-0.085999144</td>
</tr>
<tr>
<td>12.211324880</td>
<td>12.211324880</td>
<td>12.211324880</td>
</tr>
<tr>
<td>22.150635120</td>
<td>21.50635120</td>
<td>-0.523598754</td>
</tr>
<tr>
<td>0.285591150</td>
<td>0.285591150</td>
<td>0.285591150</td>
</tr>
<tr>
<td>0.285591150</td>
<td>0.285591150</td>
<td>0.285591150</td>
</tr>
<tr>
<td>12.211324880</td>
<td>12.748609832</td>
<td>13.211324880</td>
</tr>
<tr>
<td>21.150635120</td>
<td>22.081240003</td>
<td>0.461420901</td>
</tr>
<tr>
<td>-0.523598754</td>
<td>0.461420901</td>
<td>0.669396643</td>
</tr>
<tr>
<td>0.000000000</td>
<td>0.399602212</td>
<td>0.334698322</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>0.1465351279</td>
<td>0.1231823483</td>
<td>0.4175100285</td>
</tr>
<tr>
<td>0.0573681568</td>
<td>0.1166890743</td>
<td>0.293183918</td>
</tr>
<tr>
<td>0.0891669711</td>
<td>0.013932740</td>
<td>0.0264371636</td>
</tr>
<tr>
<td>12.500000000</td>
<td>11.962715048</td>
<td>11.500000000</td>
</tr>
<tr>
<td>21.650635095</td>
<td>20.720030212</td>
<td>-0.821327468</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>-0.821327468</td>
<td>-0.937891783</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>-0.711290434</td>
<td>-0.68945892</td>
</tr>
<tr>
<td>12.500000000</td>
<td>12.500000000</td>
<td>12.500000000</td>
</tr>
<tr>
<td>20.650635095</td>
<td>21.650635095</td>
<td>0.0000000000</td>
</tr>
<tr>
<td>-0.710757432</td>
<td>0.0000000000</td>
<td>-0.710757432</td>
</tr>
<tr>
<td>-0.710757432</td>
<td>0.0000000000</td>
<td>-0.710757432</td>
</tr>
<tr>
<td>12.500000000</td>
<td>13.037286958</td>
<td>13.500000000</td>
</tr>
<tr>
<td>21.650635095</td>
<td>20.720030212</td>
<td>-0.479196873</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>-0.697196873</td>
<td>-0.317104900</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>-0.414696654</td>
<td>-0.158552450</td>
</tr>
<tr>
<td>11.500000000</td>
<td>11.425430096</td>
<td>11.500000000</td>
</tr>
<tr>
<td>21.650635095</td>
<td>21.650635095</td>
<td>-0.479196841</td>
</tr>
<tr>
<td>-0.451331773</td>
<td>-0.479196841</td>
<td>-0.451331773</td>
</tr>
<tr>
<td>-0.451331773</td>
<td>-0.479196841</td>
<td>-0.451331773</td>
</tr>
<tr>
<td>12.500000000</td>
<td>12.500000000</td>
<td>12.500000000</td>
</tr>
<tr>
<td>21.650635095</td>
<td>21.650635095</td>
<td>0.0000000000</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>0.0000000000</td>
<td>0.0000000000</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>0.0000000000</td>
<td>0.0000000000</td>
</tr>
<tr>
<td>13.500000000</td>
<td>13.574569904</td>
<td>13.500000000</td>
</tr>
<tr>
<td>21.650635095</td>
<td>21.650635095</td>
<td>0.506089159</td>
</tr>
<tr>
<td>0.475339632</td>
<td>0.506089159</td>
<td>0.475339632</td>
</tr>
<tr>
<td>0.475339632</td>
<td>0.438266091</td>
<td>0.475339632</td>
</tr>
<tr>
<td>12.500000000</td>
<td>11.962715048</td>
<td>11.500000000</td>
</tr>
<tr>
<td>21.650635095</td>
<td>22.581239777</td>
<td>0.506089191</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>0.506089191</td>
<td>0.382901280</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>0.438266095</td>
<td>0.191450640</td>
</tr>
<tr>
<td>12.500000000</td>
<td>12.500000000</td>
<td>12.500000000</td>
</tr>
<tr>
<td>22.650635095</td>
<td>21.650635095</td>
<td>0.0000000000</td>
</tr>
<tr>
<td>0.7164790909</td>
<td>0.0000000000</td>
<td>0.7164790909</td>
</tr>
<tr>
<td>0.7164790909</td>
<td>0.0000000000</td>
<td>0.7164790909</td>
</tr>
<tr>
<td>12.500000000</td>
<td>13.037286952</td>
<td>13.500000000</td>
</tr>
<tr>
<td>21.650635095</td>
<td>22.581239777</td>
<td>0.821327468</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>0.821327468</td>
<td>0.937891783</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>0.711290434</td>
<td>0.469836688</td>
</tr>
<tr>
<td>0.0799786871</td>
<td>0.0907083932</td>
<td>0.1066367643</td>
</tr>
<tr>
<td>0.0666899751</td>
<td>0.0865786623</td>
<td>0.100200378</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>0.0132887121</td>
<td>0.0041295309</td>
<td>0.0066167265</td>
</tr>
<tr>
<td>12.788675120</td>
<td>12.251390168</td>
<td>11.788675120</td>
</tr>
<tr>
<td>22.150635069</td>
<td>21.220030186</td>
<td>-0.461420901</td>
</tr>
<tr>
<td>-0.523598754</td>
<td>0.523598754</td>
<td>-0.276359838</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>-0.599602212</td>
<td>-0.276359838</td>
</tr>
<tr>
<td>12.788675120</td>
<td>12.788675120</td>
<td>12.788675120</td>
</tr>
<tr>
<td>21.150635069</td>
<td>22.150635069</td>
<td>0.523598754</td>
</tr>
<tr>
<td>-0.276359838</td>
<td>0.523598754</td>
<td>-0.276359838</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>-0.276359838</td>
<td>-0.276359838</td>
</tr>
<tr>
<td>12.788675120</td>
<td>13.325960072</td>
<td>13.788675120</td>
</tr>
<tr>
<td>22.150635069</td>
<td>21.220030186</td>
<td>0.057289316</td>
</tr>
<tr>
<td>0.523598754</td>
<td>0.057289316</td>
<td>0.243383107</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>0.049614040</td>
<td>0.121691554</td>
</tr>
<tr>
<td>11.788675120</td>
<td>11.714109215</td>
<td>11.788675120</td>
</tr>
<tr>
<td>22.150635069</td>
<td>22.150635069</td>
<td>0.057289356</td>
</tr>
<tr>
<td>0.092038743</td>
<td>0.057289356</td>
<td>0.92038743</td>
</tr>
<tr>
<td>0.092038743</td>
<td>0.049614040</td>
<td>0.92038743</td>
</tr>
<tr>
<td>12.788675120</td>
<td>12.788675120</td>
<td>12.788675120</td>
</tr>
<tr>
<td>22.150635069</td>
<td>22.150635069</td>
<td>0.523598754</td>
</tr>
<tr>
<td>0.523598754</td>
<td>0.523598754</td>
<td>0.523598754</td>
</tr>
<tr>
<td>-0.094395014</td>
<td>-2.720698932</td>
<td>-3.141592521</td>
</tr>
<tr>
<td>13.788675120</td>
<td>13.863243024</td>
<td>13.788675120</td>
</tr>
<tr>
<td>22.150635069</td>
<td>22.150635069</td>
<td>0.846887027</td>
</tr>
<tr>
<td>0.829222172</td>
<td>0.846887027</td>
<td>0.829222172</td>
</tr>
<tr>
<td>0.829222172</td>
<td>0.733425717</td>
<td>0.829222172</td>
</tr>
<tr>
<td>12.788675120</td>
<td>12.251390168</td>
<td>11.788675120</td>
</tr>
<tr>
<td>22.150635069</td>
<td>23.081239952</td>
<td>0.846887045</td>
</tr>
<tr>
<td>0.523598754</td>
<td>0.846887045</td>
<td>0.774947539</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>0.733425676</td>
<td>0.387473770</td>
</tr>
<tr>
<td>12.788675120</td>
<td>12.788675120</td>
<td>12.788675120</td>
</tr>
<tr>
<td>23.150635069</td>
<td>22.150635069</td>
<td>0.966434783</td>
</tr>
<tr>
<td>0.966434783</td>
<td>0.523598754</td>
<td>0.966434783</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>0.966434783</td>
<td>0.966434783</td>
</tr>
<tr>
<td>12.788675120</td>
<td>12.788675120</td>
<td>12.788675120</td>
</tr>
<tr>
<td>22.150635069</td>
<td>23.081239952</td>
<td>1.026447805</td>
</tr>
<tr>
<td>0.523598754</td>
<td>1.026447805</td>
<td>1.096081854</td>
</tr>
<tr>
<td>0.0000000000</td>
<td>0.888929852</td>
<td>0.54804927</td>
</tr>
<tr>
<td>0.2665990866</td>
<td>0.0242907566</td>
<td>0.1519827014</td>
</tr>
<tr>
<td>0.17187704035</td>
<td>0.0135500150</td>
<td>0.1186043801</td>
</tr>
<tr>
<td>0.0947286829</td>
<td>0.0107407416</td>
<td>0.0333738212</td>
</tr>
</tbody>
</table>
C. DIGITAL FILTERING

C.1 Nth Order Differences

A 5x5 unbiased digital filter for the 8-neighbor grid lattice allows the use of third order differences in the preferential directions. The grid lattice’s window will include every pixel within two pixels of the window’s central element pixel. In general, larger digital filters allow the modeling of higher order differences.

Consider the 3rd order difference in the $\Delta U$ direction.

\[
f(3)(U) = \left\{ \left[ \begin{array}{c} \frac{f(U+2\Delta U) - f(U+\Delta U)}{\Delta U} - \frac{f(U+\Delta U) - f(U)}{\Delta U} \\ \frac{f(U) - f(U-\Delta U)}{\Delta U} - \frac{f(U-\Delta U) - f(U-2\Delta U)}{\Delta U} \end{array} \right] / \Delta U \right\} / 2\Delta U
\]

\[
= \frac{f(U+2\Delta U) - f(U+\Delta U) - f(U+\Delta U) + f(U)}{2(\Delta U)^3} + \frac{-f(U) + f(U-\Delta U) + f(U-\Delta U) - f(U-2\Delta U)}{2(\Delta U)^3}
\]

\[
= \frac{f(U+2\Delta U) - 2f(U+\Delta U) + 2f(U-\Delta U) - f(U-2\Delta U)}{2(\Delta U)^3}
\]

(C.1)

So that 3rd order difference convolution matrices can be produced by using the following vector within each
preferential direction of the digital filter.

\[
H'_{3} = \begin{bmatrix} -1 & 2 & 0 & -2 & 1 \end{bmatrix}
\]

(C.2)

All unbiased convolutions using odd-ordered \((1,3,5,...)\) differences will have zero in the digital filter’s central element. Therefore, the construction of the convolution matrix that is using odd-ordered differences is easier because the preferential directions are not sharing the central pixel in the unbiased convolution window.

A 6-neighbor grid lattice can have its elementary window expanded around the digital filter’s central matrix component, \(h(0,0)\). The window size may be specified by the maximum number of pixel steps from the window’s central element, \(n\). The total number of pixels for an expanded 6-neighbor grid lattice window is less than the total number of pixels for the comparable 8-neighbor grid lattice window, where all the window’s pixels are within \(n\) pixel steps of the window’s central element.

\[
(2n+1) + 2\left[ \sum_{i=n+1}^{2n} (i) \right] \quad ; \text{n step 6-neighbor window size}
\]

\[
(2n+1)^2 \quad ; \text{n step 8-neighbor window size}
\]

Figure C.1 illustrates a nonorthogonal address scheme for the 6-neighbor grid lattice window size of \(n=2\), where a
row, column position of index array

Figure C.1: 6-Neighbor Grid Lattice N=2 Window
skewed row and column structure is used. The maximum pixel step window size of n=2 will allow the modeling of 3rd order differences in the 6-neighbor grid lattice.

C.2 Conversion Between Cartesian Position And Array Address

Most gridded representations of the spatial plane use the 8-neighbor grid lattice. The conversion between Cartesian position, \((x,y)\), and row and column array address, \((r,c)\), is straightforward.

\[
\begin{bmatrix}
x \\
y
\end{bmatrix} = \begin{bmatrix}
\Delta X & 0 \\
0 & \Delta Y
\end{bmatrix} \begin{bmatrix}
r \\
c
\end{bmatrix} \tag{C.3a}
\]

\[
\begin{bmatrix}
r \\
c
\end{bmatrix} = \begin{bmatrix}
\Delta X & 0 \\
0 & \Delta Y
\end{bmatrix}^{-1} \begin{bmatrix}
x \\
y
\end{bmatrix} \tag{C.3b}
\]

Where \(\Delta X\) and \(\Delta Y\) are grid post spacings in the Figure 2.1 orthogonal \(x\) and \(y\) preferential directions. Conversions between address and position are just scalar multiplication operations.

The 6-neighbor grid lattice preferential directions are not orthogonal to each other. Consider the two preferential directions of \(v\) and \(w\) in Figure 2.1. The conversion between Cartesian position, \((x,y)\), and row and column array address, \((r,c)\), is an affine transformation between the \((x,y)\) and \((v,w)\) coordinate systems.
\[ \begin{bmatrix} x \\ y \end{bmatrix} = [\Delta D] \begin{bmatrix} 1 & \sin 30^\circ \\ 0 & \cos 30^\circ \end{bmatrix} \begin{bmatrix} r \\ c \end{bmatrix} \] (C.4a)

\[ \begin{bmatrix} r \\ c \end{bmatrix} = [\Delta D]^{-1} \begin{bmatrix} 1 & \sin 30^\circ \\ 0 & \cos 30^\circ \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix} \] (C.4b)

Where \( \Delta D \) is the grid post spacing distance in all the 6-neighbor grid lattice's preferential directions. So the 6-neighbor grid lattice's conversion between position and address is a skewed coordinate transformation.

Consider the 6-neighbor grid lattice storage array with the following row and column array addresses, \((r,c)\).

\[
\begin{bmatrix}
(-1,-1) & (-1, 0) & (-1, 1) \\
( 0,-1) & ( 0, 0) & ( 0, 1) \\
( 1,-1) & ( 1, 0) & ( 1, 1)
\end{bmatrix}
\] (C.5)

The array address to Cartesian position transformation for the 6-neighbor grid lattice's elementary window, where \(n=1\), creates a sparse matrix of Cartesian position, \((x,y)\).

\[
[\Delta D] \begin{bmatrix}
\text{empty} & (-1, 0) & (-\sin 30^\circ, \cos 30^\circ) \\
(-\sin 30^\circ, -\cos 30^\circ) & ( 0, 0) & ( \sin 30^\circ, \cos 30^\circ) \\
( \sin 30^\circ, -\cos 30^\circ) & ( 1, 0) & \text{empty}
\end{bmatrix}
\] (C.6)
where \( \sin 30^\circ = 1 - \sin 30^\circ \). For this paper's purpose, the 6-neighbor grid lattice's elementary window storage array diagonal directions corresponded to the \( u \) and \( w \) preferential directions.

\[
\begin{bmatrix}
(-\sin 30^\circ, -\cos 30^\circ) & (-1, 0) & (-\sin 30^\circ, \cos 30^\circ) \\
\text{empty} & (0, 0) & \text{empty} \\
(\sin 30^\circ, -\cos 30^\circ) & (1, 0) & (\sin 30^\circ, \cos 30^\circ)
\end{bmatrix}
\]

(C.7)

That gives a better intuitive sense of direction in the 6-neighbor grid lattice's 3x3 elementary window. However, the ability to maintain such intuitive connectivity is lost when the 6-neighbor grid lattice's window size increases to \( n > 1 \).

The 6-neighbor grid lattice can have the elementary window expanded farther around the elementary window's central element, so that the window size is specified in terms of maximum pixel steps, \( n \), from the central pixel, \( (0,0) \). Figure C.1 illustrates the 6-neighbor grid lattice window size of \( n=2 \). In figure C.1 the row and column array addresses, \( (r,c) \), are shown in each hexagonal pixel (hexel). A sparse 5x5 matrix is produced.
Where the pluses, +, are the only used array elements when using Equation C.4 for conversion between array address and Cartesian position.

C.3 Sample Digital Filter Matrix Convolutions

This section displays the 6-neighbor grid lattice’s digital filtering matrix convolution results along the circular edge line where the mathematical Laplacian is maximized on the radial arctangent profile forming a continuous two-dimensional mathematical surface of rotation, at the sample point where edge orientation equals 60° for the 6-neighbor grid lattice. These are numbers from the TEST.DAT file produced by write statements from the FILTER.FOR routine.
The 8-neighbor grid lattice's digital filtering matrix convolution results can be displayed at the same sample point where the 6-neighbor grid lattice results were displayed.
Both calculations illustrate the different calculations even though each digital filter’s central element is in an identical sample point position.

C.4 Radial Profiles Perpendicular To Edge Lines

The discrepancy curves, for each circular edge of constant radial distance from the surface of rotation’s radix, included critical points at different edge orientations that are indicated in Table 4.1. Each n-neighbor grid lattice’s digital Laplacian filter matrix convolution results are compared along two identical radial profiles (at different orientations) perpendicular to the edge lines on the arctangent surface of rotation. That verifies the differences in the precision of n-neighbor digital Laplacians.

Figure C.2 shows the relative precision of the 4-neighbor, 6-neighbor, and 8-neighbor digital Laplacians. Where each respective n-neighbor grid compares paired radial profile’s digital Laplacian filter matrix convolution differences. One radial profile is oriented at the minimum error curve position of the circular edge lines, and one radial profile is oriented at the maximum error curve position of the circular edge lines, positions which are indicated in Table 4.1. The width of each curve along the y-axis represents the range of different digital Laplacian matrix convolution results along identical radial profiles.
Figure C.2: Radial Profile Digital Laplacian Differences
perpendicular to the circular edge lines.

The width of each curve with respect to the vertical axis decreases as the precision of the digital Laplacian increases. The range of all n-neighbor grid lattice discrepancy curves is greatest nearer the shoulders of the arctangent profile function's radial ramp. The 6-neighbor digital Laplacian is the least wide curve, hence it is the most precise digital Laplacian.

C.5 Frequency Domain Digital Filters

The digital image can be frequency filtered in its spatial domain. That is accomplished by utilizing the row to column vector products of regular matrix multiplication.

An alternate interpretation of the matrix convolution equation can be used to frequency filter the digital image in its spatial domain.

\[
g(x,y) = \int \int h(s) f(x-s,y-t) h(t) \, ds \, dt \quad (C.9a)
\]
\[
g(x_i,y_j) = \sum_s \sum_t h_x(i,s) f(x_i+s,y_j+t) h_y(t,j) \quad (C.9b)
\]

if \( h(s,t)=h(s)h(t) \). That interpretation produces vector products between the digital image and pairs of digital filters. Where \( h_x=[h(i,0),...,h(i,m-1)] \) is the \( i \)th row vector of the \( H_x \) matrix, that will produce vector products with all the \( F \) matrix's column vectors in the image's \( x \) direction. And where \( h_y=[h(0,j),...,h(n-1,j)]^t \) is the \( j \)th
column vector of the $H_y$ matrix, that will produce vector products with all the $F$ matrix's row vectors in the image's $y$ direction. The $h_x(i,s)$ and $h_y(t,j)$ are vector elements that are discrete values of the $h(s)$ and $h(t)$ functions, respectively. So that if $F$ is a matrix of $(m,n)$ dimensions, then

$$G_{m,n} = H_x_{m,m} F_{m,n} H_y_{n,n}$$  \hspace{1cm} (C.10)

A vector $h$ is periodic if:

$$h(d) = h(d+p)$$

for all $h$ elements. The vector's periodicity is $p(\Delta U)$ where $h=[h_0, \ldots, h_{m-1}]$ is a vector of dimension $m$, with grid post spacing $\Delta U$ in the $y$ direction. The periodicity reflects the vector's frequency.

The digital image $F$ can be filtered by periodic vectors, if those vectors are contained as rows or columns of the $H$ matrices. The vector components will just be discrete values of continuous periodic functions of frequency $p(\Delta U)/(2\pi r)$; where $(\Delta U)/(2\pi r)$ is the grid post spacing of the digital image converted to radians of a circle of radius $r$, in the direction the vector products are occurring on the digital image.
The image can be filtered of certain frequencies in its spatial domain by transforming the $G$ matrix to a $G'$ matrix, accomplished by zeroing out portions of the $G$ matrix recognized as paired vector products in two directions of $F$, between vector pairs of various periodicities.

\[
\frac{F'}{m,n} = \left( \frac{H_x}{m,m} \right)^{-1} \frac{G'}{m,n} \left( \frac{H_y}{n,n} \right)^{-1}
\]  

(C.11)

if the $H$ matrices were originally designed as nonsingular matrices. Certain periodic functions do exist to produce nonsingular $H$ matrices. Examples include the Discrete Fourier Transformation, the Hadamard Transformation, and the Karhunen-Loeve Compression. The original $F$ image matrix will be reconstructed as $F'$, with certain frequencies caused by paired vector products (in the $G$ matrix) being removed from the original digital image by the inverse transformation on the $G'$ matrix.

C.6 Grid Lattice Scanner Designs

The sensor's spots are arranged in row and sample, $(r,s)$, when producing an 8-neighbor grid lattice digital image. A 6-neighbor grid lattice digital image can be produced by offsetting sensor spots along the scan direction on alternating scans. Figure C.3 illustrates resultant sensor spot positions for scanners, for both the 8-neighbor and 6-neighbor grid lattice digital images.
If the flying spot sensor(s) are attached to a wheel all the scan directions will be identical. In a push-broom scanner there can be a zig-zag pushing pattern of each row of sensors. Figure C.4 illustrates those scanner designs by showing the motion of rows of scan spots.

An 8-neighbor grid lattice scanner could be modified to a 6-neighbor grid lattice scanner by adjusting timing mechanisms controlling the scanner’s sensor array positions in the sample direction (along each scan). An adjustment of the scan stepover may be necessary to maintain the connectivity of the scan spots, comparing spot patterns of the 8-neighbor scanner versus the 6-neighbor scanner with identical spot sizes in Figure C.3.
Where the scan spot coordinates are row and sample, \((r,s)\), scan positions, indicated by the following symbols.

**Single Spot**

\(r,s\)

**Row of Spots**

8-Neighbor Grid Lattice Scanner Spot Design

\[
\begin{align*}
(r-1) & : s-2 \quad s-1 \quad s \quad s+1 \quad s+2 \\
(r) & : s+2 \quad s+1 \quad s \quad s-1 \quad s-2 \\
(r+1) & : s-2 \quad s-1 \quad s \quad s+1 \quad s+2
\end{align*}
\]

6-Neighbor Grid Lattice Scanner Spot Design

\[
\begin{align*}
(r-1) & : s-2 \quad s-1 \quad s \quad s+1 \quad s+2 \\
(r) & : s+2 \quad s+1 \quad s \quad s-1 \quad s-2 \\
(r+1) & : s-2 \quad s-1 \quad s \quad s+1 \quad s+2
\end{align*}
\]

Figure C.3: Scanner Spot Designs
Where each row of scan spots is a solid bar. And the scan direction is indicated by arrows.

**Flying Spot(s)**

---

**Push Broom (pushed pair of offset sensor rows)**

---

**Push Broom (zig-zag pushing pattern of single sensor row)**

---

Figure C.4: 6-Neighbor Grid Lattice Scanning Patterns
Roger Brown was born in Ketchikan, Alaska on 30 June 1955. He graduated from the University of Oregon in June 1978, where he received a Bachelor of Science degree with a Mathematics major and an Economics major. He was hired by the Defense Mapping Agency Hydrographic Topographic Center (DMAHTC) as a Mathematician/Programmer in February 1980. His duties included application software maintenance for photogrammetry systems. He attended the DMAHTC Long Term Full Time Training (LTFTT) program from October 1985 through September 1986, where he did graduate level course work in Geodetic Science (Photogrammetry and Geodesy) at Virginia Polytechnic Institute’s (VPI) Northern Virginia Graduate Center. After returning from LTFTT, he was reassigned as a Cartographer where he worked for divisions of the Digital Products Department and the Systems Center. He currently is assigned to the Digital Products Department Production Support Office’s Engineering and Modernization Branch (DPP2), where his duties include automated data processing techniques and engineering work on several photogrammetry and cartography systems. He received a Master of Science degree in Civil Engineering from VPI in May 1989. He currently lives in Gaithersburg, Maryland with his wife Robina and their three children (Tyler, Alicia, and Adrian).