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 Kottman 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.

ACKNOWLEDGEMENTS

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
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 and our children for allowing me to concentrate on studies while attending the program.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

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 tesselated 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 tesselation 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 quantitity 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^{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,

$\text{grad}(f)=\nabla f=(\partial f/\partial x)\underline{i}+(\partial f/\partial y)\underline{j}=[\partial f/\partial x,\partial f/\partial y]$.

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 emmittance properties within the image. Rosenfeld & Kak (1982) present various brightness distribution properties including gray level variance, co-occurance 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 similiar 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.

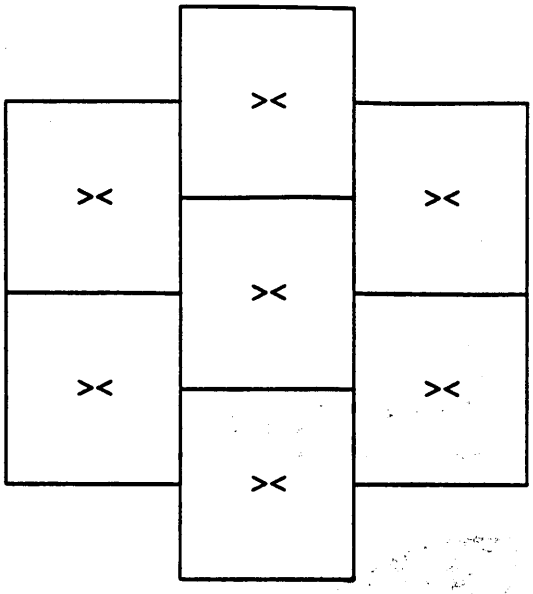
An elementary window of the grid lattice is defined in terms of one central grid post and a certain number of that post's nearest neighboring posts. An n-neighbor grid lattice is an elementary window of n+1 posts. An elementary window is large enough to explain the whole grid lattice's geometry.

Processing of the grid lattice's digital data requires that the grid post patterns are described by an array

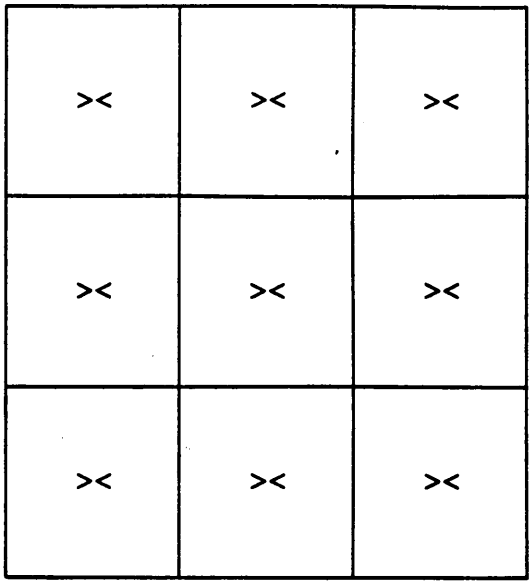
storage function. Index arrays may contain displacement vectors that describe the positional relationship between the grid lattice window's central post and each of its neighboring posts within the elementary window. The grid post spacing distance is the displacement vector's magnitude. Preferential 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.

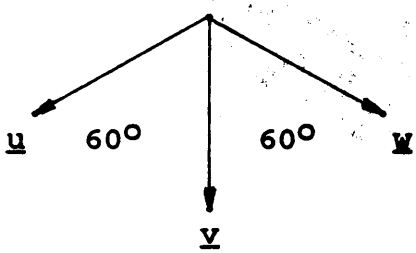
Table 2.1 shows the displacement vectors for each grid lattice. The 8-neighbor grid lattice's elementary window is a rectangular array of nine posts that yield four displacement vectors we shall label as $\underline{x}$, $\underline{y}$, $\underline{s}$, $\underline{t}$. The 6-neighbor grid lattice's elementary window is a hexagonal array of seven posts that yield three displacement vectors we shall label as $\underline{u}$, $\underline{v}$, $\underline{w}$. The 4-neighbor grid lattice is a special case of the 8-neighbor grid lattice without the $\underline{s}$ and $\underline{t}$ 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.



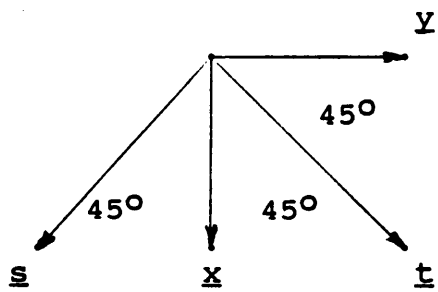
6-Neighbor Grid



8-Neighbor Grid



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

$\underline{x} = [\ \Delta X\ ,\ 0.0\ ]$

$\underline{y} = [\ 0.0\ ,\ \Delta Y\ ]$

$\underline{s} = [\ \Delta X\ ,\ -\Delta Y\ ]$

$\underline{t} = [\ \Delta X\ ,\ \Delta Y\ ]$

6-Neighbor Grid Lattice

$\underline{u} = [\ (\Delta X)\sin(30^{o})\ ,\ -(\Delta X)\cos(30^{o})\ ]$

$\underline{v} = [\ \Delta X\ ,\ 0.0\ ]$

$\underline{w} = [\ (\Delta X)\sin(30^{o})\ ,\ (\Delta X)\cos(30^{o})\ ]$

4-Neighbor Grid Lattice

$\underline{x} = [\ \Delta X\ ,\ 0.0\ ]$

$\underline{y} = [\ 0.0\ ,\ \Delta Y\ ]$

Note: ΔX and ΔY are displacements in the coordinate system's orthogonal directions, $\underline{x}$ and $\underline{y}$ 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

$$\Delta' X = (\Delta X)\sin(30^{o}) = (\Delta X)/2$$

$$\Delta' Y = (\Delta X)\cos(30^{o})$$

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

4-Neighbor Index Array

$$\begin{bmatrix} \text{empty} & (-\Delta X, 0.0) & \text{empty} \\ (0.0, -\Delta Y) & (0.0, 0.0) & (0.0, \Delta Y) \\ \text{empty} & (\Delta X, 0.0) & \text{empty} \end{bmatrix}$$

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^{o} for the 4-neighbor grid lattice, 60^{o} for the 6-neighbor grid lattice, and 45^{o} 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 $\underline{u}$ or $\underline{w}$ displacement vector elements, $(\Delta'X, \Delta'Y)$, in the displacement vector index array is

$$|\Delta'Y|/|\Delta'X| = \tan 60^{o} = (3)^{1/2} \qquad (2.1)$$

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. ΔX does not equal ΔY), 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 posess 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 imgery. Less imagery distortions will improve the imagery analysis quality, particulary 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 anomolies 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

$$\frac{\delta f(u)}{\delta u} @(U) = \lim_{\Delta U \to {}^{+}0} \frac{f(U+\Delta U) - f(U)}{\Delta U} \qquad (3.1a)$$

$$= \lim_{\Delta U \to {}^{-}0} \frac{f(U) - f(U-\Delta U)}{\Delta U}$$

$$\frac{\partial f(x,y)}{\partial x} @(X,Y) = \lim_{\Delta X \to {}^{+}0} \frac{f(X+\Delta X,Y) - f(X,Y)}{\Delta X}$$

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

$$\frac{\partial f(x,y)}{\partial y} @(X,Y) = \lim_{\Delta Y \to {}^{+}0} \frac{f(X,Y+\Delta Y) - f(X,Y)}{\Delta Y} \qquad (3.1c)$$

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

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 ΔX, ΔY, and ΔU spatial distances in the $\underline{x}$, $\underline{y}$, and $\underline{u}$ directions. The grid post spacings of ΔX, ΔY, or ΔU 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 ΔX and ΔY orthogonal directions, and at position (u) for the ΔU direction.

$$f^{(1)}(U) = \frac{f(U+\Delta U) - f(U-\Delta U)}{2\Delta U} \qquad (3.2a)$$

$$f_X^{(1)}(X,Y) = \frac{f(X+\Delta X,Y) - f(X-\Delta X,Y)}{2\Delta X} \qquad (3.2b)$$

$$f_Y^{(1)}(X,Y) = \frac{f(X,Y+\Delta Y) - f(X,Y-\Delta Y)}{2\Delta Y} \qquad (3.2c)$$

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

$$= \frac{f(U+\Delta U)}{(\Delta U)^2} - \frac{2\ f(U)}{(\Delta U)^2} + \frac{f(U-\Delta U)}{(\Delta U)^2} \qquad (3.3a)$$

$$f_X^{(2)}(X,Y) = \frac{f(X+\Delta X,Y)}{(\Delta X)^2} - \frac{2\ f(X,Y)}{(\Delta X)^2} + \frac{f(X-\Delta X,Y)}{(\Delta X)^2} \qquad (3.3b)$$

$$f_Y^{(2)}(X,Y) = \frac{f(X,Y+\Delta Y)}{(\Delta Y)^2} - \frac{2\ f(X,Y)}{(\Delta Y)^2} + \frac{f(X,Y-\Delta Y)}{(\Delta Y)^2} \qquad (3.3c)$$

The substitutions for ΔU 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.

$h(s,t)$ = point spread sampling function

$f(x,y)$ = function surface to be sampled

$g(x,y)$ = filtered surface of sample values

$$g(x,y) = \int\int f(x,y)\, h(x-s,y-t)\, \delta s\, \delta t$$
$$= \int\int f(x-s,y-t)\, h(s,t)\, \delta s\, \delta t \qquad (3.4)$$

$$g(X_i,Y_j) = \sum_t \sum_s f(X_{i-s},Y_{j-t})\, h(s,t) \qquad (3.5)$$

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 $\underline{F}$, $\underline{G}$, and $\underline{H}$ are matrices whose elements are discrete values of the f, g, and h functions respectively. $\underline{H}$ is called the digital filter. Each resultant element of the $\underline{G}$ matrix is the summation of matrix element products between the $\underline{F}$ and $\underline{H}$ matrices, which we indicate by

$$g(X_i,Y_j) = \underset{r,c}{\underline{F'}} * \underset{r,c}{\underline{H}} \qquad (3.6)$$

The convolution window's dimensions are specified as row by columns, (r,c). $\underline{H}$ is a matrix of row by column dimensions. The $\underline{F}'$ matrix is a row by columns partition of the $\underline{F}$ matrix. In Equations 3.5 and 3.6, $\underline{H}$ is being convolved on $\underline{F}$. Each h(s,t) element of the $\underline{H}$ matrix will be multiplied by each corresponding $f(X_{i-s},Y_{j-t})$ element of the $\underline{F}'$ 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 $\underline{G}$ 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

$$g(X_i,Y_j) = \underset{3,3}{\underline{F'}} * \underset{3,3}{\underline{H}} \qquad (3.7a)$$

$$\underset{3,3}{F'} = \begin{bmatrix} f(X_{i-1},Y_{j-1}) & f(X_{i-1},Y_j) & f(X_{i-1},Y_{j+1}) \\ f(X_i,Y_{j-1}) & f(X_i,Y_j) & f(X_i,Y_{j+1}) \\ f(X_{i+1},Y_{j-1}) & f(X_{i+1},Y_j) & f(X_{i+1},Y_{j+1}) \end{bmatrix} \quad (3.7b)$$

$$\underset{3,3}{H} = \begin{bmatrix} h(1,1) & h(1,0) & h(1,-1) \\ h(0,1) & h(0,0) & h(0,-1) \\ h(-1,1) & h(-1,0) & h(-1,-1) \end{bmatrix} \quad (3.7c)$$

where $X_{i+k}=X+k\Delta X$ and $Y_{j+k}=Y+k\Delta Y$ for $k=-1,0,1$.

A directional ambiguity is inherent in the mathematical definition of convolution because of the 180^{o} rotation of the point spread sampling function. A switch of direction will be introduced deliberately into the $\underline{H}$ matrix while using matrix convolution to estimate derivatives by differences. In Equation 3.5 substitutions are made for the $\underline{H}$ 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 $\underline{H}$, anyway.

$$\underset{3,3}{\underline{H}} = \begin{bmatrix} h(-1,-1) & h(-1,\ 0) & h(-1,\ 1) \\ h(\ 0,-1) & h(\ 0,\ 0) & h(\ 0,\ 1) \\ h(\ 1,-1) & h(\ 1,\ 0) & h(\ 1,\ 1) \end{bmatrix} \qquad (3.7d)$$

Unbiased differencing operations result in a symmetric digital filter, $\underline{H}$, where the digital filter's central element is h(0,0). Matrix convolution is symmetric when the (2m+1) by (2n+1) dimensioned $\underline{H}$ matrix elements are h(s,t), where s=(-m,...,-1,0,1,...m) and t=(-n,...,-1,0,1,...,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 $\underline{x}$ and $\underline{y}$ directions, we obtain the following:

$$\underset{3,3}{\underline{H}}_x = \begin{bmatrix} 0 & (\Delta X)^{-2} & 0 \\ 0 & -2(\Delta X)^{-2} & 0 \\ 0 & (\Delta X)^{-2} & 0 \end{bmatrix} \qquad (3.8a)$$

$$\underset{3,3}{\underline{H}}_y = \begin{bmatrix} 0 & 0 & 0 \\ (\Delta Y)^{-2} & -2(\Delta Y)^{-2} & (\Delta Y)^{-2} \\ 0 & 0 & 0 \end{bmatrix} \qquad (3.8b)$$

The $\underline{H}_x$ matrix's middle column vector is identical to the $\underline{H}_y$ 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 the $\underline{H}$ matrix according to grid post spacing distance.

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

$$\underline{H} = \underline{H}' \text{ o } \underline{W} \tag{3.9}$$

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

$$\underset{3,3}{H}x = \underset{3,3}{H'}x \circ \underset{3,3}{W}x \qquad (3.10a)$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \circ \begin{bmatrix} 0 & (\Delta X)^{-2} & 0 \\ 0 & (\Delta X)^{-2} & 0 \\ 0 & (\Delta X)^{-2} & 0 \end{bmatrix}$$

$$\underset{3,3}{H}y = \underset{3,3}{H'}y \circ \underset{3,3}{W}y \qquad (3.10b)$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} \circ \begin{bmatrix} 0 & 0 & 0 \\ (\Delta Y)^{-2} & (\Delta Y)^{-2} & (\Delta Y)^{-2} \\ 0 & 0 & 0 \end{bmatrix}$$

One may notice that the $\underset{3,3}{H}$ 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

$$g(U_i) = \sum_s f(U_{i-s})\, h(s) \qquad (3.11)$$
$$= f(U_{i-1})h(-1) + f(U_i)h(0) + f(U_{i+1})h(1)$$

The i^{th}-order difference digital filter is the $\underline{H}_i=[h(-1),h(0),h(1)]$ vector. The digital filters are

$$\underset{1,3}{\underline{H}}_0 = [\ 0 \quad 1 \quad 0\ ] = f(u) \qquad (3.12)$$

$$\underset{1,3}{\underline{H}}_1 = \underline{H'}_1 \circ \underline{W}_1 \qquad (3.13)$$
$$= [\ -1 \quad 0 \quad 1\ ] \circ [\ 1/2 \quad 0 \quad 1/2\ ]$$
$$= [\ -1/2 \quad 0 \quad 1/2\ ]$$

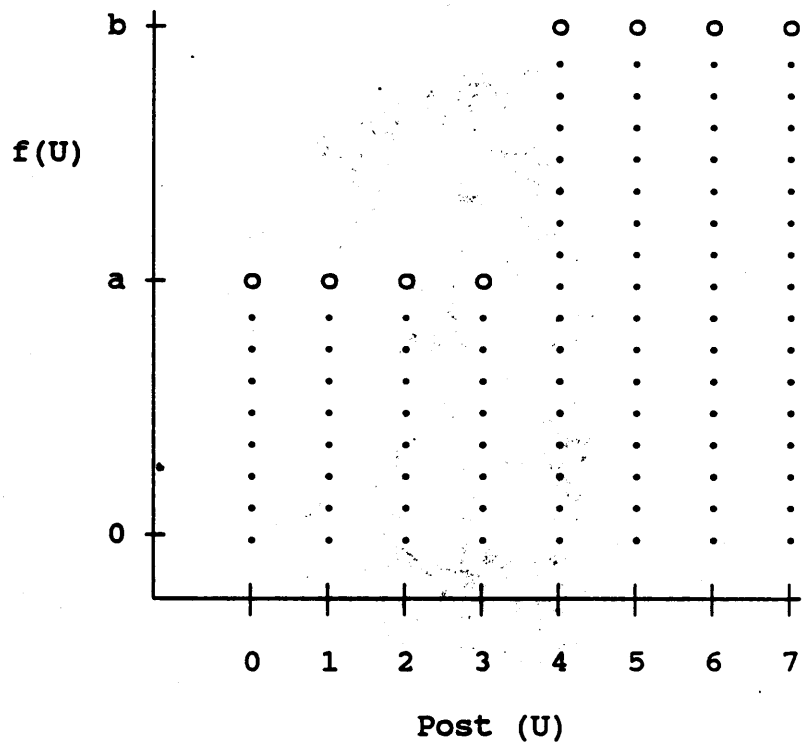


Figure 3.1: Digital Step Edge

$$\underset{1,3}{\underline{H}}_2 = \underline{H'}_2 \circ \underline{W}_2 \qquad (3.14)$$

$$= [\ 1\ \ -2\ \ 1\ ] \circ [\ 1\ \ 1\ \ 1\ ]$$

$$= [\ 1\ \ -2\ \ 1\ ]$$

where $\Delta U=1$ is assumed to be unit distance grid post spacing in the $\underline{u}$ 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 $\underline{H}_1=[-1,1,0]$ or $\underline{H}_1=[0,-1,1]$, 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

$$\underline{G}_1 = \underline{F'} * \underline{H}_1$$

$$= [\ a\ a\ a\ a\ b\ b\ b\ b\ ] * [\ -1/2\ \ 0\ \ 1/2\ ]$$

$$= [\ ?\ \ 0\ \ 0\ \ (b-a)/2\ \ (b-a)/2\ \ 0\ \ 0\ \ ?\ ] \qquad (3.15)$$

$$\underline{G}_2 = \underline{F'} * \underline{H}_2$$

$$= [\ a\ a\ a\ a\ b\ b\ b\ b\ ] * [\ 1\ -2\ \ 1\ ]$$

$$= [\ ?\ \ 0\ \ 0\ (b-a)\ \ (a-b)\ \ 0\ \ 0\ \ ?\ ] \qquad (3.16)$$

Figure 3.2 displays the convolution results. The '?' elements of each vector are indeterminant unless the ends of $\underline{F'}$ are extended to accomodate the span of the digital filter. On the step edge, the magnitudes of the non-zero

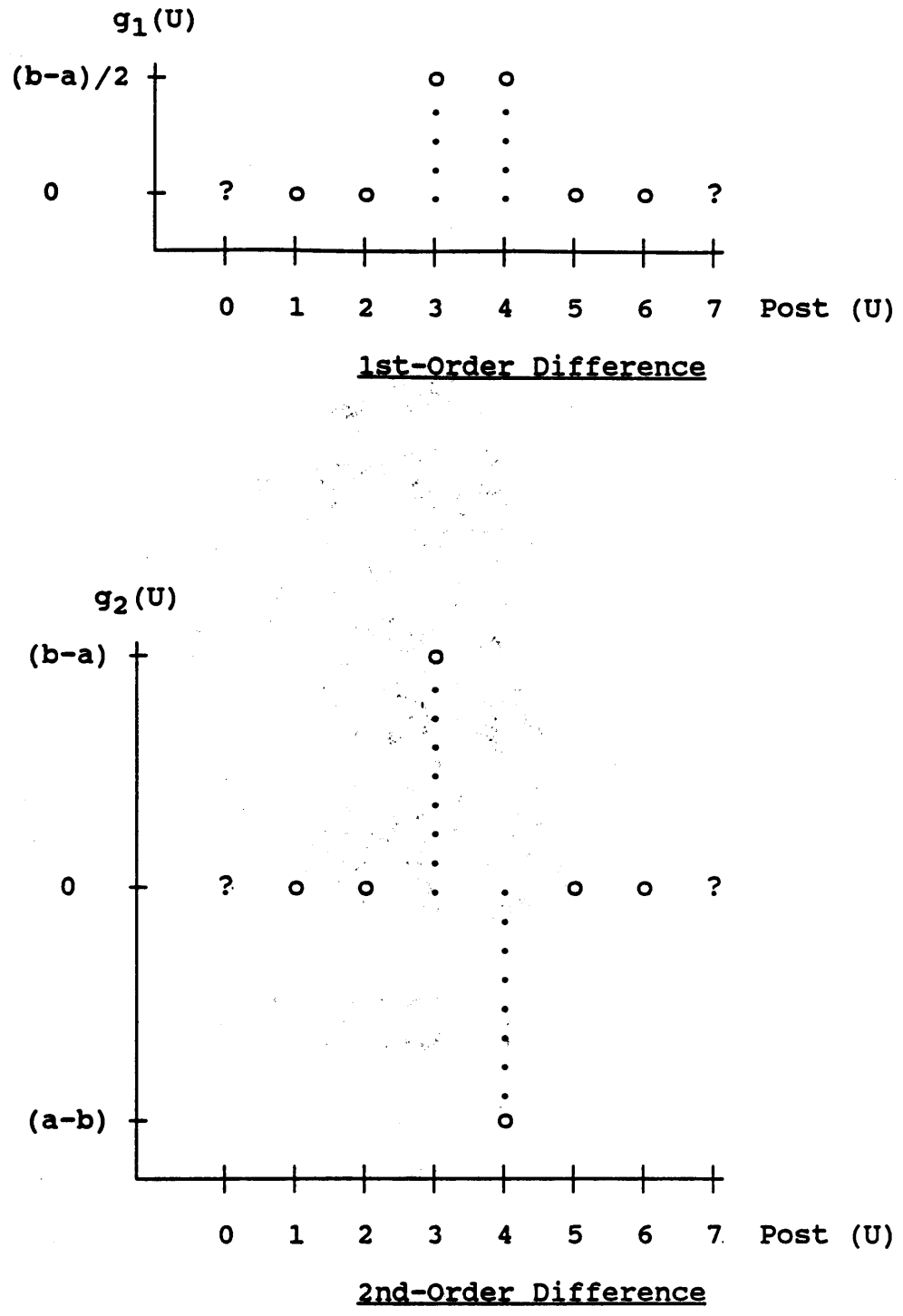


Figure 3.2: Filtered Digital Step Edge

elements for $\underline{G}_2$ are $2\Delta U$ times greater than the magnitudes of the non-zero elements of $\underline{G}_1$.

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.

$$f^{(0)}(u) = v \tan^{-1}(h u) \tag{3.17a}$$

$$f^{(1)}(u) = vh / (1 + h^2u^2) \tag{3.17b}$$

$$[f^{(1)}(u)]^2 = v^2h^2 / (1 + h^2u^2)^2 \tag{3.17c}$$

$$\begin{aligned} f^{(2)}(u) &= (vh)(-1)(2uh^2)/(1 + h^2u^2)^2 \\ &= (-2uhh^2v^2/v)/(1 + h^2u^2)^2 \\ &= (-2uh/v)[f^{(1)}(u)]^2 \end{aligned} \tag{3.17d}$$

$$\begin{aligned} f^{(3)}(u) &= (-2h/v)[f^{(1)}(u)]^2 + \\ &\quad (-2uh/v)(2)[f^{(1)}(u)][f^{(2)}(u)] \\ &= (-2h/v)[f^{(1)}(u)]^2 + \\ &\quad (-2uh/v)(2)[f^{(1)}(u)]\{(-2uh/v)[f^{(1)}(u)]^2\} \\ &= (-2h/v)[f^{(1)}(u)]^2 + \\ &\quad (-2uh/v)(2)(-2uh/v)[f^{(1)}(u)]^3 \\ &= (-2h/v)[f^{(1)}(u)]^2\{1 - (4u^2h/v)[f^{(1)}(u)]\} \\ &= (-2h/v)[f^{(1)}(u)]^2\{1 - (4u^2h^2)/(1 + h^2u^2)\} \end{aligned} \tag{3.17e}$$

where $u=x-x_o$ is the distance from the center of the ramp, x_o, and f(u) is brightness at position u.

The center of the ramp, x_o, is the arctangent profile's inflection point. The developed arctangent function's derivatives include horizonal 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_o. 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_o, 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.

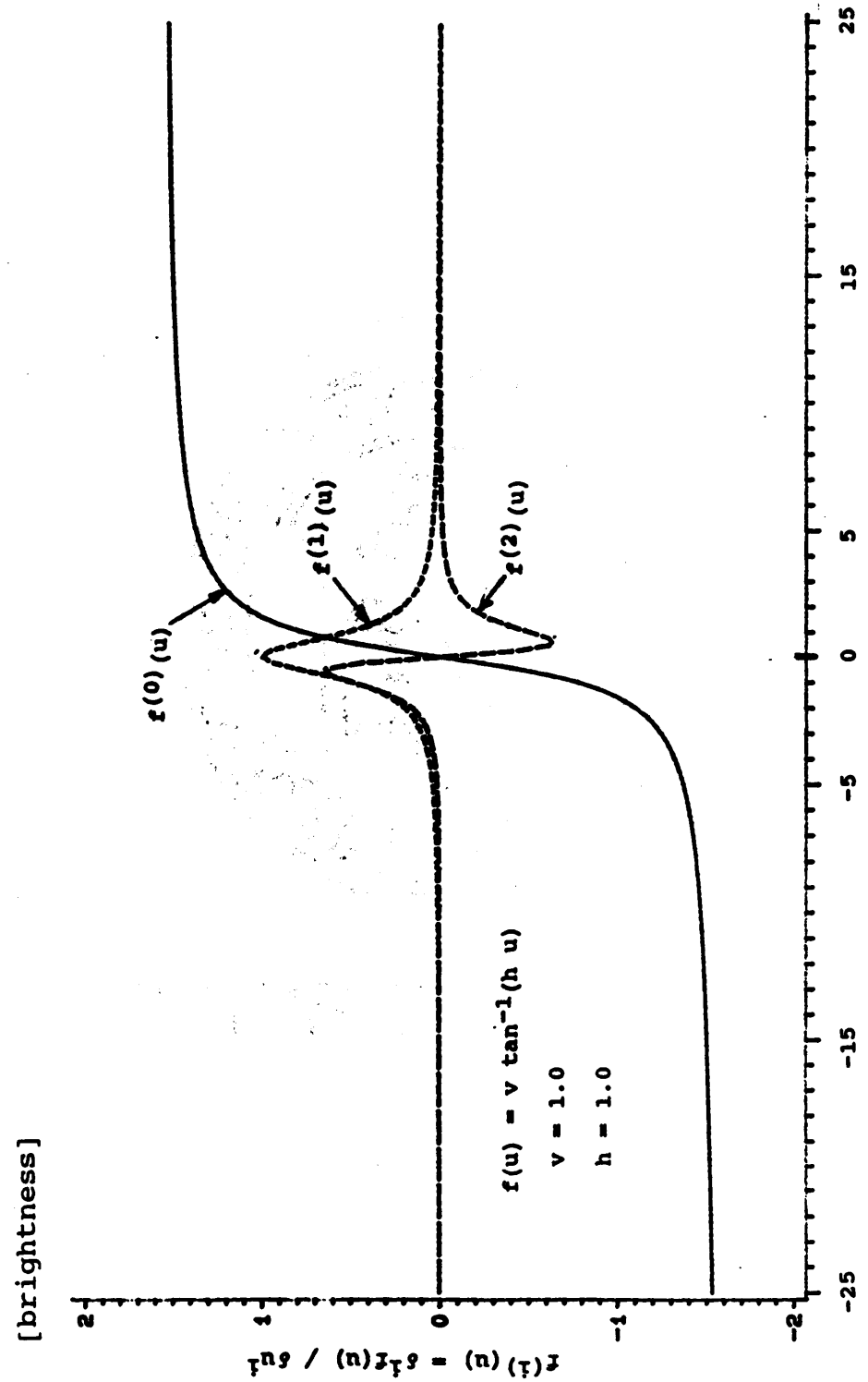


Figure 3.3: Continuous Ramp Arctangent Derivatives

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 ΔU for slope calculations using the second-order difference filter is possible because the filter is inherently unbiased. A smaller ΔU should be preferable when the profile is unpredictable because $\Delta f(U)/\Delta U$ only approaches $\delta f(u)/\delta u$ as ΔU 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_o, Y_o), in the spatial domain. Let the surface of rotation formulations be

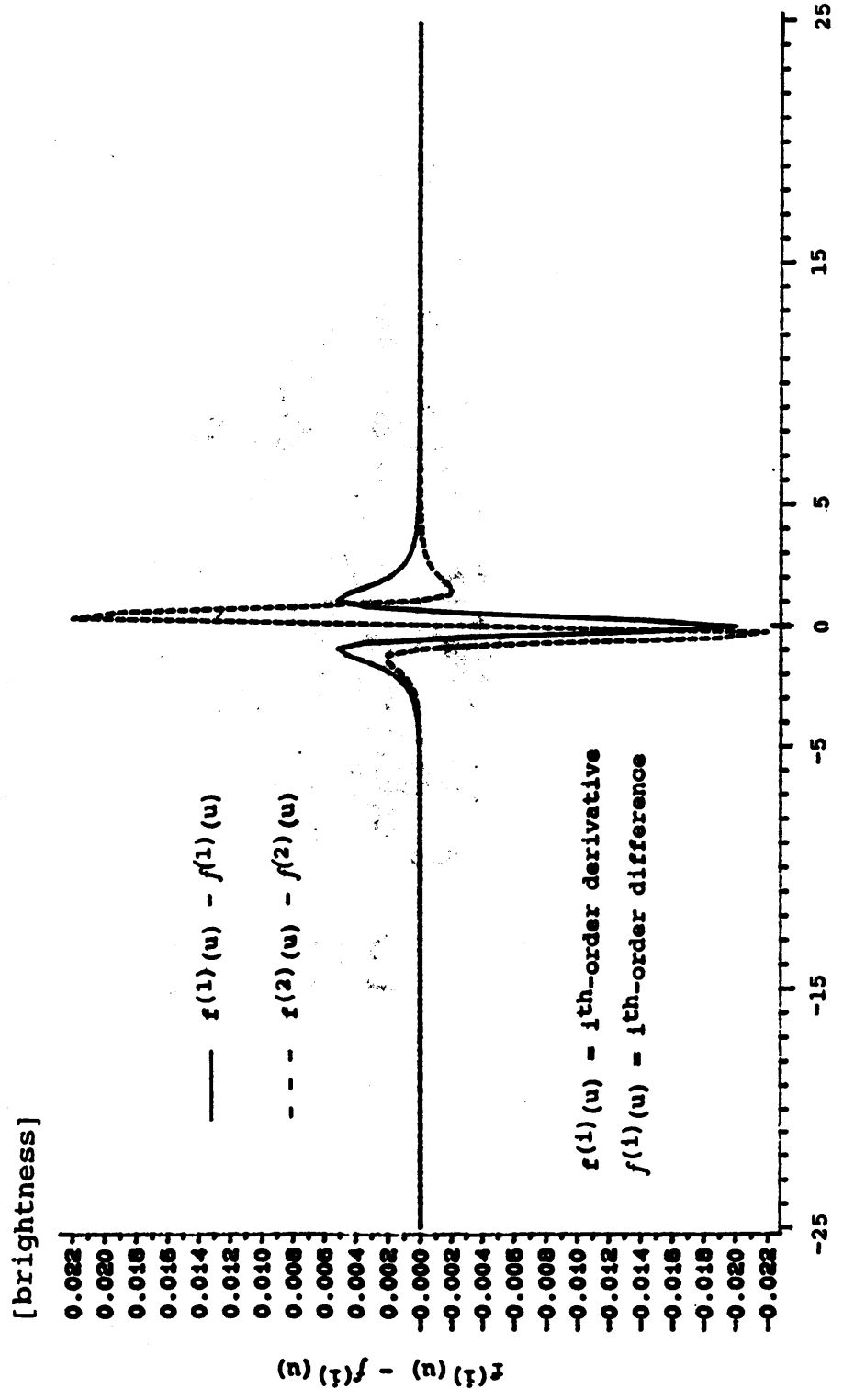


Figure 3.4: Arctangent Derivatives Versus Differences

$$x' = X - X_o$$

$$y' = Y - Y_o$$

$$R = [\ x'^2 + y'^2\ ]^{1/2}$$

Then

$$f(x',y') = f(R) \qquad (3.18)$$

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

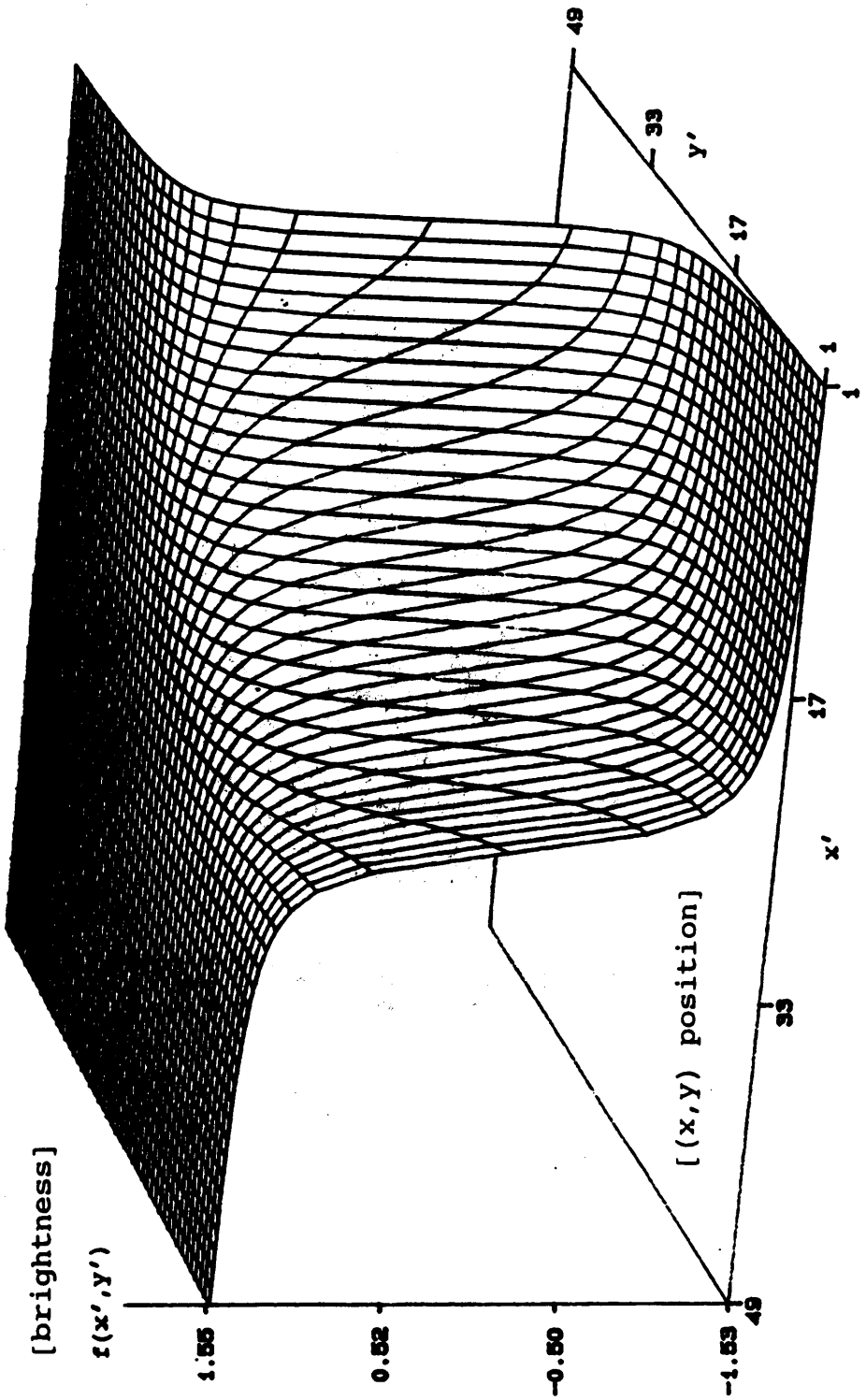


Figure 3.5: Radial Arctangent Profile's Surface of Rotation

that is on the edge's line. The direction of the radius of curvature vector, that we shall label as $\underline{n}$, is determined from the vector cross product of the tangent vectors in the $\underline{x}$ and $\underline{y}$ directions. The direction of $\underline{n}$ is the edge line's orientation at that (x,y) point. Let

$$z = f(x,y)$$

Then direction of the tangents in the $\underline{x}$ and $\underline{y}$ direction are

$$\tau_x = \tan^{-1}(\partial z/\partial x)$$
$$\tau_y = \tan^{-1}(\partial z/\partial y)$$

and the radius of curvature vector is

$$\begin{aligned}\underline{n} = &[\ \underline{i}(\cos\tau_x) + \underline{j}(0) + \underline{k}(\sin\tau_x)\ ] \\ &\qquad (X)\quad ;\ \text{vector cross-product} \\ &[\ \underline{i}(0) + \underline{j}(\cos\tau_y) + \underline{k}(\sin\tau_y)\ ] \\ = &[\ -\underline{i}(\sin\tau_x)(\cos\tau_y) - \underline{j}(\cos\tau_x)(\sin\tau_y) \\ &\qquad + \underline{k}(\cos\tau_x)(\cos\tau_y)\ ] \end{aligned} \qquad (3.19)$$

The edge's orienation in the (x,y) spatial domain is labeled as θ. The edge's orientation is calculated by ignoring the $\underline{k}$ element of the $\underline{n}$ vector, and then taking the arctangent of the ratio of the $\underline{i}$ and $\underline{j}$ elements of the $\underline{n}$ vector, so that

$$\theta = \tan^{-1}\left[\frac{(\cos \tau_x)(\sin \tau_y)}{(\sin \tau_x)(\cos \tau_y)} \right]$$

$$= \tan^{-1}[\ (\text{cotan}\ \tau_x)(\tan \tau_y)\ ]$$

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

$$= \tan^{-1}[\ (\partial z/\partial y)\ /\ (\partial z/\partial x)\ ] \qquad (3.20)$$

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 $(\partial x,\partial y,\partial z)$, respectively.

The edge sharpness may be described by the derivatives along the one-dimensional brightness profile corresponding to the direction, θ, of the perpendicular radius of curvature vector, $\underline{n}$. 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.

$$\mathcal{L}[f(x,y)] = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$
$$= f_x^{(2)}(x,y) + f_y^{(2)}(x,y) \qquad (3.21)$$

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.

$$\begin{aligned} \mathcal{L}[f(x',y')] &= \frac{\partial^2 f(x',y')}{\partial x'^2} + \frac{\partial^2 f(x',y')}{\partial y'^2} \\ &= f^{(2)}(R) - f^{(1)}(R)/R \\ &= \frac{\delta^2 f(R)}{\delta R^2} - \frac{\delta f(R)}{\delta R} / R \end{aligned} \qquad (3.22)$$

The Laplacian is a derivative operator that will give a circular edge line a constant value because $\mathcal{L}[f(x',y']$ 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 ΔX and ΔY 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).

$$\mathcal{L}[f(x,y)] \approx g(X_i,Y_j) \tag{3.23}$$

$$= \frac{F'}{3,3} * \frac{H}{3,3}x + \frac{F'}{3,3} * \frac{H}{3,3}y$$

$$= \frac{F'}{3,3} * \left(\frac{H}{3,3}x + \frac{H}{3,3}y \right)$$

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

$$= \frac{F'}{3,3} * \frac{H}{3,3}4$$

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.

$$\frac{H'_4}{3,3} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -2+-2 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.24}$$

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 $\underline{x}$ and $\underline{y}$.

$$\begin{aligned}
\mathcal{L}[f(x,y)] &\sim g(X_i,Y_j) \\
&= [\ f(X_i,Y_{j-1}) + f(X_{i-1},Y_j) + \\
&\quad f(X_{i+1},Y_j) + f(X_i,Y_{j+1})\ ] - \\
&\qquad 4f(X_i,Y_j) \\
&= f(X_i,Y_{j-1}) - f(X_i,Y_j) + \\
&\quad f(X_{i-1},Y_j) - f(X_i,Y_j) + \\
&\quad f(X_{i+1},Y_j) - f(X_i,Y_j) + \\
&\quad f(X_i,Y_{j+1}) - f(X_i,Y_j)
\end{aligned} \tag{3.25}$$

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

$$\Sigma_t \Sigma_s\ h(s,t) = 0 \tag{3.26}$$

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, $\underline{H}'$, and their respective weight matrices, $\underline{W}$, in terms of grid post spacing distances in the orthogonal ΔX and ΔY 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, $\underline{W}_4$ and $\underline{W}_8$. And the $\underline{H}'_6$ identity digital Laplacian filter was weighted with grid post spacing density equivalent to $\underline{H}_8$.

$$\underline{H}_4 = \underline{H}'_4 \circ \underline{W}_4$$

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

$$= \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.27}$$

Table 3.1a: 4-Neighbor Digital Laplacian Filter Matrices

Filter Matrix

$$\underset{3,3}{H'_4} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Weight Matrix

$$\underset{3,3}{W_4} = \begin{bmatrix} 0 & (\Delta X)^{-2} & 0 \\ (\Delta Y)^{-2} & w & (\Delta Y)^{-2} \\ 0 & (\Delta X)^{-2} & 0 \end{bmatrix}$$

$$w = [\ 2(\Delta X)^{-2} + 2(\Delta Y)^{-2}\ ] / 4$$

Table 3.1b: 6-Neighbor Digital Laplacian Filter Matrices

Filter Matrix

$$\underset{3,3}{H'_6} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & -6 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Weight Matrix

$$\underset{3,3}{W_6} = \begin{bmatrix} (\Delta')^{-2} & (\Delta X)^{-2} & (\Delta')^{-2} \\ 0 & w & 0 \\ (\Delta')^{-2} & (\Delta X)^{-2} & (\Delta')^{-2} \end{bmatrix}$$

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

$$w = [2(\Delta X)^{-2} + 4(\Delta')^{-2}] / 6$$

Table 3.1c: 8-Neighbor Digital Laplacian Filter Matrices

Filter Matrix

$$\underset{3,3}{H'_8} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Weight Matrix

$$\underset{3,3}{W_8} = \begin{bmatrix} (\Delta')^{-2} & (\Delta X)^{-2} & (\Delta')^{-2} \\ (\Delta Y)^{-2} & w & (\Delta Y)^{-2} \\ (\Delta')^{-2} & (\Delta X)^{-2} & (\Delta')^{-2} \end{bmatrix}$$

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

$$w = [\ 2(\Delta X)^{-2} + 2(\Delta Y)^{-2} + 4(\Delta')^{-2}\ ] / 8$$

$$\underline{H}_6 = \underline{H}'_6 \circ \underline{W}_6$$

$$= \begin{bmatrix} 1 & 1 & 1 \\ x & -6 & x \\ 1 & 1 & 1 \end{bmatrix} \circ \begin{bmatrix} (3/4)^{1/2} & (3/4)^{1/2} & (3/4)^{1/2} \\ x & (3/4)^{1/2} & x \\ (3/4)^{1/2} & (3/4)^{1/2} & (3/4)^{1/2} \end{bmatrix}$$

$$= [3^{1/2}] \begin{bmatrix} 1/2 & 1/2 & 1/2 \\ x & -3 & x \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \qquad (3.28)$$

$$\underline{H}_8 = \underline{H}'_8 \circ \underline{W}_8$$

$$= \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \circ \begin{bmatrix} 1/2 & 1 & 1/2 \\ 1 & 3/4 & 1 \\ 1/2 & 1 & 1/2 \end{bmatrix}$$

$$= \begin{bmatrix} 1/2 & 1 & 1/2 \\ 1 & -6 & 1 \\ 1/2 & 1 & 1/2 \end{bmatrix} \qquad (3.29)$$

The 'x' matrix elements are unused thus making $\underline{H}_6$ a sparse matrix. The $\underline{H}_6$ matrix's diagonal directions correspond to the 6-neighbor grid lattice elementary window's $\underline{u}$ and $\underline{w}$ preferential directions (see Figure 2.1).

All three weighted digital Laplacian filters ($\underline{H}_4$, $\underline{H}_6$, and $\underline{H}_8$) 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.

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

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

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

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 occuring 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 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, θ, 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 $\mathcal{L}[f(x',y')]$ only equals a

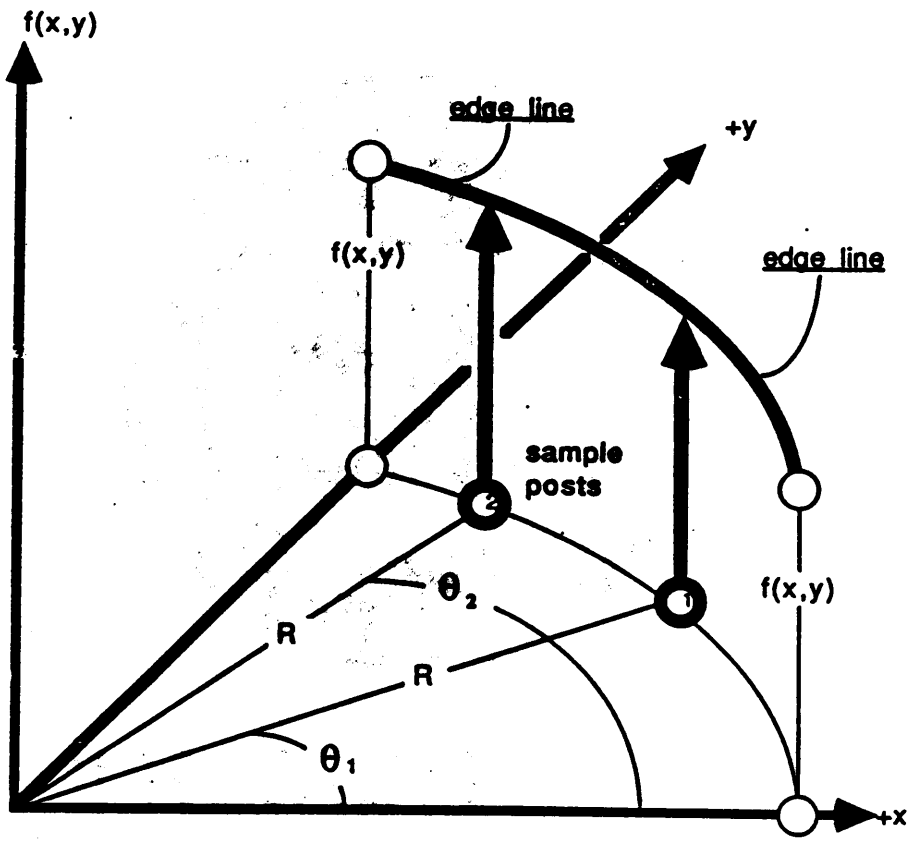


Figure 4.1: Circular Edge Line Sample Post Orientation

function of the edge line's radius of curvature, $R=[x'^2+y'^2]^{1/2}$. 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, $\kappa=1/R$, and the mathematical surface of rotation's first-order differential.

$$[f^{(2)}(U)]-[f^{(2)}(R)] = [f^{(1)}(R)]/R$$
$$= \kappa[f^{(1)}(R)] \qquad (4.1)$$

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 $R_{max}=R-[(3^{1/2})h]^{-1}$, $R_{mid}=R$, and $R_{min}=R+[(3^{1/2})h]^{-1}$. The radii's subscript (max, mid, 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 ΔX and ΔY 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 ΔX and ΔY 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 accomodate 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

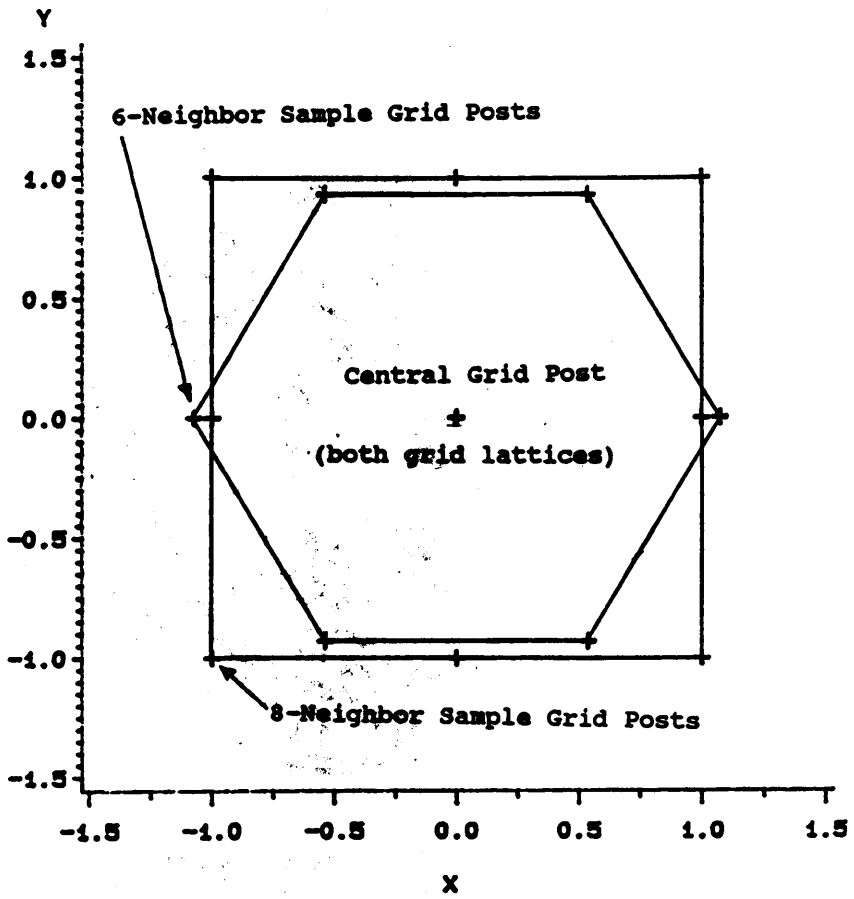


Figure 4.2:

6-Neighbor And 8-Neighbor Grid's Elementary Window

for each grid lattice will determine the prefered 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_{max}, R_{mid}, R_{min}) with sample points having different edge orientations, θ, 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^{o} to 360^{o}.

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 θ degrees on each discrepenancy curve's x-axis. We denote the mathematical Laplacian as $\mathcal{L}[f(R)]$, 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 $\mathcal{L}[f(x',y')]=\mathcal{L}[f(R)]$ 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, θ, on the x-axis. Since R is constant on the circular edge line but the digital Laplacian is not constant, 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.

$$\xi_n(R,\theta) = \left| \mathcal{L}[f(R)] - g_n(R\cos\theta,R\sin\theta) \right| \qquad (4.4)$$

Figure 4.3 displays the discrepancy curves of mathematical Laplacian minus the digital Laplacian for three different

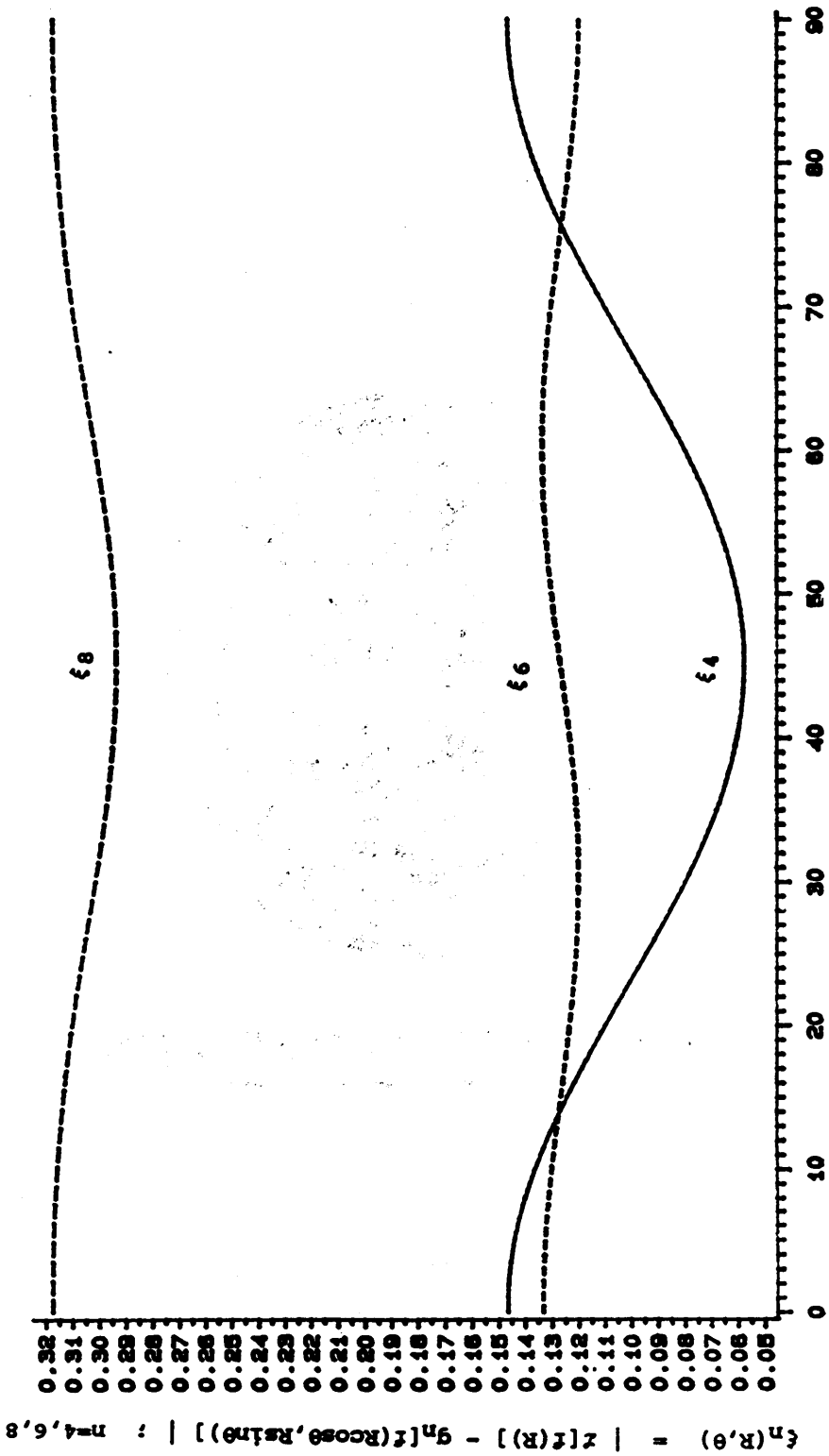


Figure 4.3a: R_{max} Digital Laplacian Discrepancy Curves

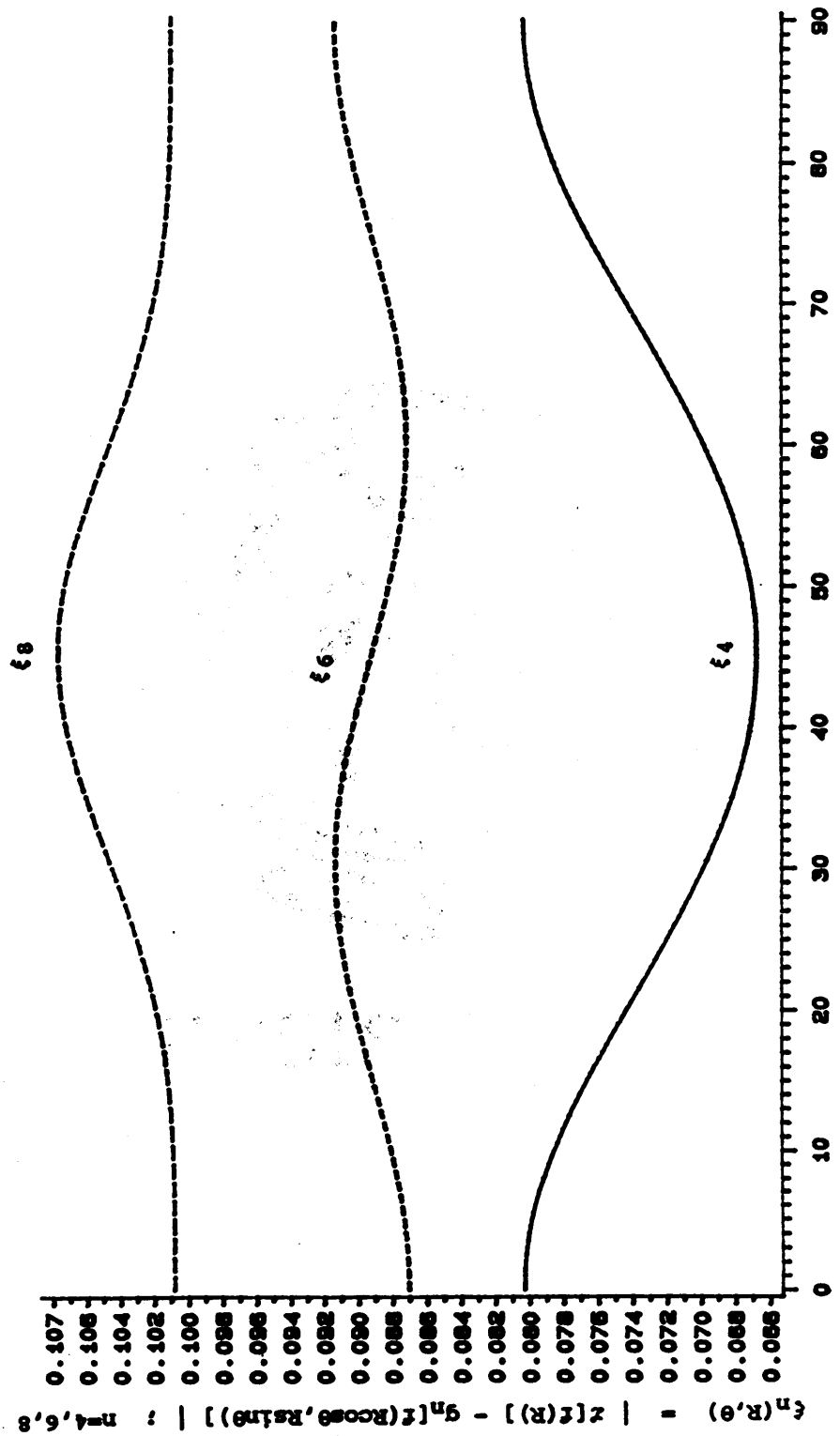


Figure 4.3b: R_{mid} Digital Laplacian Discrepancy Curves

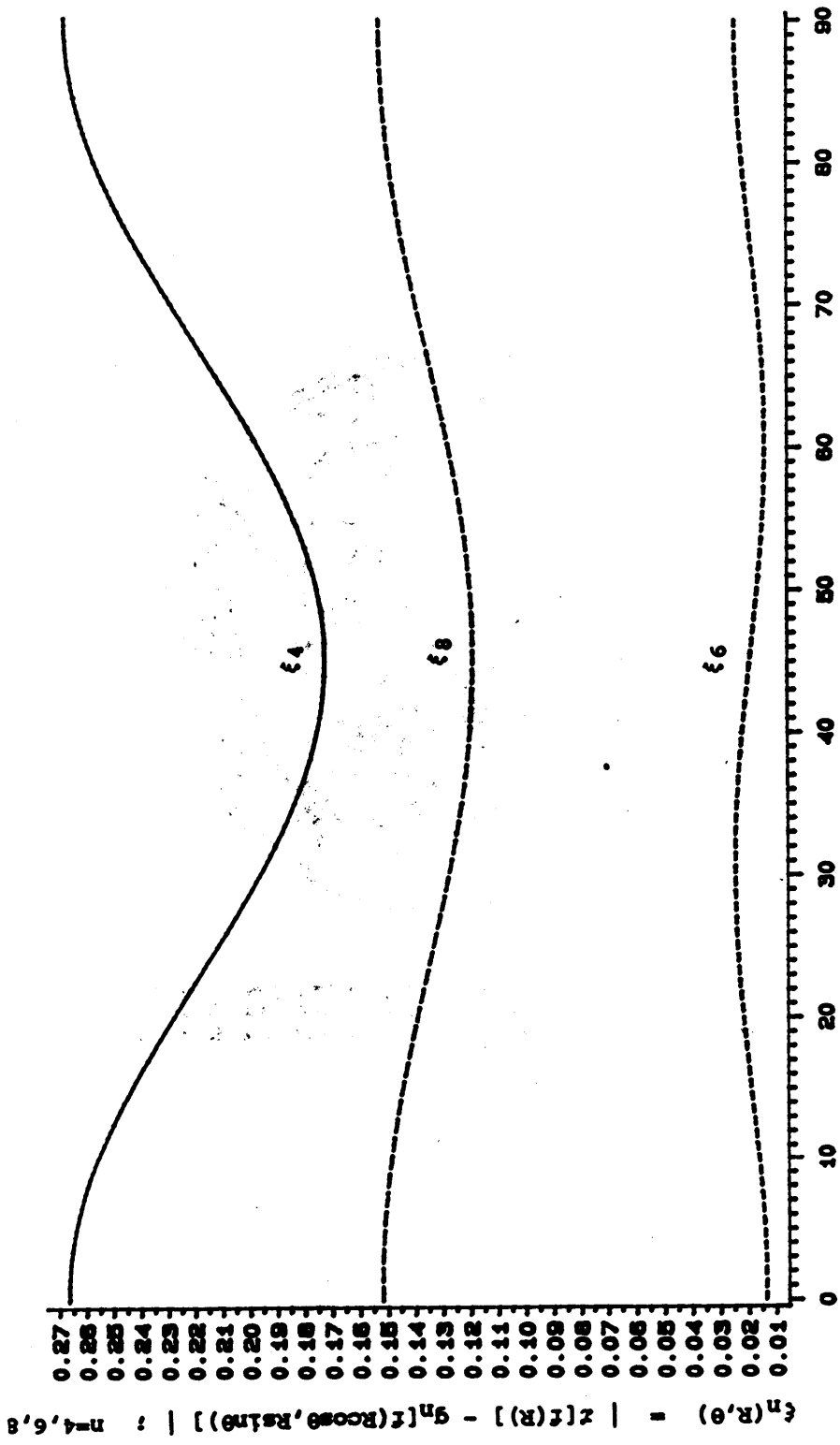


Figure 4.3c: R_{min} Digital Laplacian Discrepancy Curves

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 $R_{max}=25-[3^{1/2}]^{-1}$, $R_{mid}=25$, and $R_{min}=25+[3^{1/2}]^{-1}$. 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 hense 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

	R_{max}	R_{mid}	R_{min}		
Filter	Amplitude			Periodicity	Critical Points
4	0.089	0.013	0.095	$\pi/2$	$0^o, 45^o, 90^o$
6	0.013	0.004	0.011	$\pi/3$	$0^o, 30^o, 60^o, 90^o$
8	0.024	0.007	0.033	$\pi/2$	$0^o, 45^o, 90^o$
	Constant Value				
$\mathcal{L}$	0.619	-0.040	-0.679		

The discrepancy curve amplitudes for the circular edge lines of R_{max} and R_{min} are a smaller proportion of each edge line's mathematical Laplacian than for R_{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 ($\underline{s}$ and $\underline{t}$ compared to $\underline{x}$ and $\underline{y}$) to gain the precision of the 6-neighbor grid with all displacement vectors ($\underline{u}$, $\underline{v}$, $\underline{w}$) 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^{o} 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

Billingsly, Fred. "Digital Processing and Reprocessing." Manual Of Remote Sensing. American Society of Photogrammetry, 1983.

Castleman, Kenneth. Digital Image Processing. New Jersey: Prentice-Hall, 1979.

Crettez, Jean-Pier. "A Pseudo-Cosign Transform for Hexagonal Tesselation with an Heptarchical Origination." 5th International Conference On Pattern Recognition, IEEE Computer Society Proceedings. Institute of Electrical & Electronics Engineers, 1980.

Davies, David and Bouldin, Donald. "An Edge Operator Without Directional Preference." Pattern Recongnition and Image Processing, IEEE Computer Society Proceedings. Institute of Electrical & Electronic Engineers, 1978.

Fairchild, Dana. "The Effect of Enumeration Unit Shape on Isopleth Map Accuracy." Technical Papers Of The American Congress On Surveying And Mapping. American Congress on Surveying and Mapping (ACSM), 1981.

MacEachren, Alan. "Thematic Map Accuracy: The Influence of Enumeration Unit Size and Compactness." Technical Papers Of The American Congress On Surveying And Mapping. American Congress on Surveying and Mapping (ACSM), 1982.

Muehrcke, Phillip. Map Use. Wisconsin: JP Publications, 1978.

Kresig, Erwin. Advanced Engineering Mathematics. New York: John Wiley, 1983

Larson, Roland and Hostetler, Robert. Calculus. Massachusets: D.C. Heath and Company, 1979.

Peterson, Ivars. "Tiling To Infinity". Science News, 16 July 1988.

Rosenfeld, Azriel and Kak, Avinash. Digital Image Processing. Florida: Academic Press Inc., 1982.

Rucker, Rudy. Mind Tools: The Five Levels of Mathematical Reality. Boston: Houghton Mifflin Company, 1987.

SAS/GRAPH User's Guide, Version 5 Edition. North Carolina: SAS Institute Incprporated, 1985.

Van-Roessal, Jan. "Conversion Of Cartesian Coordinates From And To Generalized Balanced Ternary Addresses". Photogrammetric Engineering And Remote Sensing. American Society For Photogrammetry And Remote Sensing, November 1988.

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, $[f^{(i)}(x,y)]^j$ (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,...).

An example of property one is the Laplacian:

$$\mathcal{L}[f(x,y)] = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \qquad (A.2)$$

An example of property two is the squared magnitude of the gradient.

$$M^2 = [\partial f(x,y)/\partial x]^2 + [\partial f(x,y)/\partial y]^2 \qquad (A.3)$$

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 θ.

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

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

That means

$$\begin{aligned} x &= x' \cos(\theta) - y' \sin(\theta) \\ y &= x' \sin(\theta) + y' \cos(\theta) \\ x' &= x \cos(\theta) + y \sin(\theta) \\ y' &= -x \sin(\theta) + y \cos(\theta) \end{aligned} \qquad (A.4)$$

That produces the equalities of the partial derivatives

$$\partial x/\partial x' = \partial y/\partial y' = \partial x'/\partial x = \partial y'/\partial y = \cos(\theta)$$

$$\partial x/\partial y' = \partial y'/\partial x = -\sin(\theta)$$

$$\partial y/\partial x' = \partial x'/\partial y = \sin(\theta) \qquad \text{(A.5)}$$

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

$$f = f(x,y)$$

$$f' = f(x',y')$$

$$\frac{\partial f}{\partial x'} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial x'} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial x'}$$

$$= \frac{\partial f}{\partial x}\cos(\theta) + \frac{\partial f}{\partial y}\sin(\theta)$$

$$\frac{\partial f}{\partial y'} = \frac{\partial f}{\partial y}\frac{\partial y}{\partial y'} + \frac{\partial f}{\partial x}\frac{\partial x}{\partial y'}$$

$$= \frac{\partial f}{\partial y}\cos(\theta) - \frac{\partial f}{\partial x}\sin(\theta)$$

$$\frac{\partial f}{\partial x} = \frac{\partial f'}{\partial x'}\frac{\partial x'}{\partial x} + \frac{\partial f'}{\partial y'}\frac{\partial y'}{\partial x}$$

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

$$\frac{\partial f}{\partial y} = \frac{\partial f'}{\partial y'}\frac{\partial y'}{\partial y} + \frac{\partial f'}{\partial x'}\frac{\partial x'}{\partial y}$$

$$= \frac{\partial f'}{\partial y'}\cos(\theta) + \frac{\partial f'}{\partial x'}\sin(\theta) \qquad (A.7)$$

The Laplacian, $\mathcal{L}[f(x,y)]=\partial^2 f/\partial x^2+\partial^2 f/\partial^2 y$, 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 $\mathcal{L}[f(x,y)]=\mathcal{L}[f(x',y')]$, using the chain rule between the rotated coordinate systems.

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \frac{\partial}{\partial x'}\frac{\partial x'}{\partial x}\frac{\partial f}{\partial x} + \frac{\partial}{\partial y'}\frac{\partial y'}{\partial x}\frac{\partial f}{\partial x}$$

$$+ \frac{\partial}{\partial x'}\cdot\frac{\partial x'}{\partial y}\frac{\partial f}{\partial y} + \frac{\partial}{\partial y'}\frac{\partial y'}{\partial y}\frac{\partial f}{\partial y}$$

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

$$+ \frac{\partial}{\partial x'}\frac{\partial f}{\partial y}\sin(\theta) + \frac{\partial}{\partial y'}\frac{\partial f}{\partial y}\cos(\theta)$$

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

$$- \frac{\partial}{\partial y'}\left[\frac{\partial f'}{\partial x'}\cos(\theta) - \frac{\partial f'}{\partial y'}\sin(\theta)\right]\sin(\theta)$$

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

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

$$= \frac{\partial}{\partial x'}\frac{\partial f'}{\partial x'}\cos^2(\theta) - \frac{\partial}{\partial x'}\frac{\partial f'}{\partial y'}[\cos(\theta)\sin(\theta)]$$

$$+ \frac{\partial}{\partial y'}\frac{\partial f'}{\partial y'}\sin^2(\theta) - \frac{\partial}{\partial y'}\frac{\partial f'}{\partial x'}[\cos(\theta)\sin(\theta)]$$

$$+ \frac{\partial}{\partial x'}\frac{\partial f'}{\partial x'}\sin^2(\theta) + \frac{\partial}{\partial x'}\frac{\partial f'}{\partial y'}[\cos(\theta)\sin(\theta)]$$

$$+ \frac{\partial}{\partial y'}\frac{\partial f'}{\partial y'}\cos^2(\theta) + \frac{\partial}{\partial y'}\frac{\partial f'}{\partial x'}[\cos(\theta)\sin(\theta)]$$

$$= \frac{\partial^2 f'}{\partial x'^2} + \frac{\partial^2 f'}{\partial y'^2} \qquad \text{(A.9)}$$

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.

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

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

$$= \lim_{\Delta U \to 0} \frac{f(U+\Delta U)}{(\Delta U)^2} + \frac{f(U-\Delta U)}{(\Delta U)^2} - \frac{2\, f(U)}{(\Delta U)^2} \qquad \text{(A.13b)}$$

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

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

$$= f_x{}^2(x,y) + f_y{}^2(x,y)$$

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

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

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

$$\approx \frac{f(X+\Delta X,Y)}{(\Delta X)^2} + \frac{f(X-\Delta X,Y)}{(\Delta X)^2} - \frac{2\ f(X,Y)}{(\Delta X)^2} + \frac{f(X,Y+\Delta Y)}{(\Delta Y)^2} + \frac{f(X,Y-\Delta Y)}{(\Delta Y)^2} - \frac{2\ f(X,Y)}{(\Delta Y)^2} \qquad \text{(A.14)}$$

$$= f_x^{(2)}(x,y) + f_y^{(2)}(x,y)$$

The difference formulas, $f^{(i)}$, 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, $\underline{W}$, while the numerators are located in the identity matrix, $\underline{H}'$. The discrete image is located in the $\underline{F}$ matrix. Matrix convolution of the equation $\sum_t \sum_s f(X_{i-s}, Y_{j-t}) h(s,t)$ 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 $\underline{H}_1$ and $\underline{H}_2$ 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, $\underline{H}_1(0,0)$,

to be coincident with the convolution window's sample post where the first-order difference is being assigned, to cause an unbiased filter.

$$\frac{\delta f(u)}{\delta u} = \lim_{2\Delta U \to 0} \frac{f(U + \Delta U) - f(U - \Delta U)}{2\Delta U} \qquad (A.15)$$

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

$$\underset{1,3}{H} = \underset{1,3}{H'} \circ \underset{1,3}{W}$$

$$= [\ -1 \quad 0 \quad 1\ ] \circ [\ (2\Delta U)^{-1} \quad 0 \quad (2\Delta U)^{-1}\ ] \qquad (A.16)$$

The $\underline{H}$ matrix of 3x3 dimensions takes the sum of second-order differences in the orthogonal ΔX and ΔY directions.

$$\underset{3,3}{H} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \circ \begin{bmatrix} 0 & (\Delta X)^{-2} & 0 \\ 0 & (\Delta X)^{-2} & 0 \\ 0 & (\Delta X)^{-2} & 0 \end{bmatrix} +$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} \circ \begin{bmatrix} 0 & 0 & 0 \\ (\Delta Y)^{-2} & (\Delta Y)^{-2} & (\Delta Y)^{-2} \\ 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \circ$$

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

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

$$\begin{bmatrix} 0 & (\Delta X)^{-1} & 0 \\ (\Delta Y)^{-2} & [\, 2(\Delta X)^{-2} + 2(\Delta Y)^{-2} \,]/4 & (\Delta Y)^{-2} \\ 0 & (\Delta X)^{-1} & 0 \end{bmatrix}$$

(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 $\underline{H}_4$, $\underline{H}'_4$ and $\underline{W}_4$ matrices when $\Delta X=\Delta Y=1$. All the n-neighbor grid lattice equivalent digital Laplacian filters, $\underline{H}_n$, are designed so that $\underline{H}'_n$ has the sum of differences property in all preferential directions. Then each weight matrix, $\underline{W}_n$, is designed so that $\underline{H}_n$ retains the matrix elements sum equals zero property of $\underline{H}'_n$. Those properties cause $\underline{H}_4$ to equal $\underline{H}'_4$ for just the 4-neighbor grid lattice, but the $\underline{H}_n=\underline{H}'_n$ equality does not exist for $\underline{H}_6$ and $\underline{H}_8$.

$$\underline{H}_{3,3} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \circ \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \underline{H}'_{3,3} \qquad \text{(A.18)}$$

A.3 Surface of Rotation Differentials

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

$$\mathcal{L}[f(x,y)] = (\partial^2/\partial x^2)[f^{(0)}(R)] + (\partial^2/\partial y^2)[f^{(0)}(R)]$$

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

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

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

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

where

$$R = (x^2 + y^2)^{1/2}$$

$$\partial R/\partial x = x/R$$

$$\partial R/\partial y = y/R$$

$$\partial^2 R/\partial x^2 = R^{-1} - (xR^{-2})(xR^{-1}) = R^{-1} - x^2R^{-3}$$

$$\partial^2 R/\partial y^2 = R^{-1} - (yR^{-2})(yR^{-1}) = R^{-1} - y^2R^{-3}$$

$$(\partial^2 R/\partial x^2) + (\partial^2 R/\partial y^2) = 2R^{-1} - (x^2+y^2)R^{-3} = R^{-1}$$

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:

$$\{ 1 - (4h^2u^2)/(1+h^2u^2) \} = 0$$

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 $\underline{x}$ 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, $\underline{x}$ and $\underline{y}$.

$$X_6Y_6 = X_8Y_8 \qquad (A.20)$$

The constraints of equal preferential direction magnitudes for each grid lattice are

$$Y_6 = X_6 \cos(30^o) \tag{A.21a}$$

$$Y_8 = X_8 \tag{A.21b}$$

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

$$\begin{aligned} X_6 &= X_8 Y_8 \,/\, Y_6 \\ &= X_8^2 \,/\, [X_6 \cos(30^o)] \end{aligned} \tag{A.22}$$

After multiplying Equation A.22 by X_6, you get

$$\begin{aligned} X_6 &= [\, X_8^2 \,/\, \cos(30^o)\,]^{1/2} \\ &= (4/3)^{1/4}\, X_8 \\ &\approx (1.07457)\, X_8 \end{aligned} \tag{A.23}$$

Deriving Equation A.24 similarly, you get

$$\begin{aligned} Y_6 &= X_8 Y_8 \,/\, X_6 \\ &= (3/4)^{1/4}\, X_8 \\ &\approx (0.93060)\, X_8 \end{aligned} \tag{A.24}$$

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

8-Neighbor Grid		6-Neighbor Grid	
X_8	Y_8	X_6	Y_6
1.0000	1.0000	1.0746	0.9306
0.5000	0.5000	0.5373	0.4653
0.2500	0.2500	0.2686	0.2327

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 precedural 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.EXEC)
*
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
      IMPLICIT REAL (A-Z)
C
      INTEGER I,LUN/1/
      REAL XO/100.0/,H/1.0/,V/1.0/
      REAL DX/0.25/
      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 WEIT01(-1:1),WEIT02(-1:1)
```

```
      DATA FILT01/ -1 ,  0 , +1 /
      DATA FILT02/ +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/(SQRT(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) * FILT01(J) * WEIT01(J)
            SUM2 = SUM2 + Z(I+J) * FILT02(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)')
     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,'(F7.2,5F13.8)')
     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 Z*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.COPY" -----------------
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            Y   WW CC EE
C                SW SS SE
C
      COMMON /FILT/INDX04,INDX06,INDX08,
     *             FILT04,FILT06,FILT08,
     *             WEIT04,WEIT06,WEIT08
C
      INTEGER X/1/ , NW/1/,NN/4/,NE/7/ ,
     *        Y/2/ , WW/2/,CC/5/,EE/8/ ,
     *               SW/3/,SS/6/,SE/9/
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
      COMMON /SPAN/DX04,DY04,DX06,DY06
      REAL*8 DX04,DY04,DX06,DY06
C
C --- EDGE SHAPE PARAMETERS ---
C
      COMMON /SHAP/RO,H,V
      REAL*8 RO,H,V
C=====================================================================
```

B.2.5 SURFER.FOR

```
      PROGRAM SURFER
C
C---------------------------------------------------------------------
C --- DRIVER FOR GRID EDGE DETECTION SOFTWARE ------------------------
C---------------------------------------------------------------------
C ROGER BROWN , JUNE 1987
C---------------------------------------------------------------------
C
      INCLUDE (GRIDDS)
      REAL*8 ZERO/0.0D0/
      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
      R0=25.0
      H=1.0
      V=1.0
C
C --- DISPLAY MATHEMATICAL SURFACE GRID ---
C
      WRITE(1,'(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=1,49
            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 G3D 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 = R0 + (J-1)*RADJ
         CALL SURF( RAD , ZERO , Z0 , ML )
C
         WRITE(4,'(A,I1,A)')' DATA P',J,' ;'
         WRITE(4,*)'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,*)'CARDS ;'
C
```

```
      DO 20 ANG=0,90
C
         RANG = ANG * R2D
         XO = RAD*DCOS(RANG)
         YO = RAD*DSIN(RANG)
         S=ANG
         CALL FILTER( XO,YO , 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 ;'
```

```
      WRITE(4,*)'SYMBOL6 C=BLACK I=SPLINE L=3 V=NONE ;'
      WRITE(4,*)'SYMBOL7 C=BLACK I=SPLINE L=1 V=NONE ;'
      WRITE(4,*)'SYMBOL8 C=BLACK I=SPLINE L=2 V=NONE ;'
      WRITE(4,*)'SYMBOL9 C=BLACK I=SPLINE L=3 V=NONE ;'
C
      WRITE(4,*)'PROC GPLOT DATA=P3 ;'
      WRITE(4,*)'PLOT L40*ANG0 L60*ANG0 L80*ANG0 / OVERLAY ;'
      WRITE(4,*)'PLOT L41*ANG1 L61*ANG1 L81*ANG1 / OVERLAY ;'
      WRITE(4,*)'PLOT L42*ANG2 L62*ANG2 L82*ANG2 / OVERLAY ;'

      STOP
      END
```

B.2.6 CNVLTN.FOR

```
      SUBROUTINE CNVLTN
C
C-----------------------------------------------------------------------
C --- CONSTRUCT DIGITAL CONVOLUTION RELEVANT ARRAYS ------------------
C-----------------------------------------------------------------------
C ROGER BROWN , JUNE 1987
C-----------------------------------------------------------------------
C
      INCLUDE (GRIDDS)
C
      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
C --- POST SPACING ---
C
      CALL DENSIT
C
C --- 4-NEIGHBOR SPATIAL INDEX ---
C
C              | 0.0 , 0.0  ;  -DX , 0.0  ;  0.0 , 0.0  |
C     INDX() = | 0.0 , -DY  ;  0.0 , 0.0  ;  0.0 , +DY  |
```

```
C             |  0.0 , 0.0  ;  +DX , 0.0  ;  0.0 , 0.0  |
C
      INDX04(Y,WW) = -DY04
      INDX04(X,NN) = -DX04
      INDX04(X,SS) = +DX04
      INDX04(Y,EE) = +DY04
C
C --- 4-NEIGHBOR DIGITAL FILTER ---
C
C             |  0.0  ,  1.0  ,  0.0  |
C     FILT() = |  1.0  , -4.0  ,  1.0  |
C             |  0.0  ,  1.0  ,  0.0  |
C
      FILT04(WW) =  1.0
      FILT04(NN) =  1.0
      FILT04(CC) = -4.0
      FILT04(SS) =  1.0
      FILT04(EE) =  1.0
C
C --- 4-NEIGHBOR INVERSE DISTANCE WEIGHT MATRIX ---
C
C             |    0.0     1/DX**2      0.0     |
C     WEIT() = |  1/DY**2    .....     1/DY**2   |
C             |    0.0     1/DX**2      0.0     |
C
C      ..... = SUM NON-ZERO WEIT04() ELEMENTS THEN DIVIDE BY
C              ABSOLUTE VALUE OF FILT04(CC)
C
      WEIT04(WW) = 1/DY04**2
      WEIT04(NN) = 1/DX04**2
      WEIT04(SS) = 1/DX04**2
      WEIT04(EE) = 1/DY04**2
      WEIT04(CC) = ( 2/DX04**2 + 2/DY04**2 ) / 4
C
C --- 6-NEIGHBOR SPATIAL INDEX ---
C
C             | -DX/2, -DY  ;  -DX , 0.0  ; -DX/2, +DY  |
C     INDX() = |  0.0 , 0.0  ;  0.0 , 0.0  ;  0.0 , 0.0  |
C             | +DX/2, -DY  ;  +DX , 0.0  ; +DX/2, +DY  |
C
      HDX = 0.5 * DX06
      INDX06(X,NW) = -HDX
      INDX06(Y,NW) = -DY06
      INDX06(X,SW) = +HDX
      INDX06(Y,SW) = -DY06
      INDX06(X,NN) = -DX06
      INDX06(X,SS) = +DX06
      INDX06(X,NE) = -HDX
      INDX06(Y,NE) = +DY06
      INDX06(X,SE) = +HDX
      INDX06(Y,SE) = +DY06
```

```
C
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
      FILT06(WW) =  0.0
      FILT06(CC) = -6.0
      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
      DI = DSQRT( HDX*HDX + DY06*DY06 )
      WEIT06(NW) = 1/DI**2
      WEIT06(SW) = 1/DI**2
      WEIT06(NN) = 1/DX06**2
      WEIT06(SS) = 1/DX06**2
      WEIT06(NE) = 1/DI**2
      WEIT06(SE) = 1/DI**2
      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
      INDX08(X,NW) = -DX04
      INDX08(Y,NW) = -DY04
      INDX08(Y,WW) = -DY04
      INDX08(X,SW) = +DX04
      INDX08(Y,SW) = -DY04
      INDX08(X,NN) = -DX04
      INDX08(X,SS) = +DX04
      INDX08(X,NE) = -DX04
      INDX08(Y,NE) = +DY04
      INDX08(Y,EE) = +DY04
      INDX08(X,SE) = +DX04
      INDX08(Y,SE) = +DY04
C
C --- 8-NEIGHBOR DIGITAL FILTER ---
C
C             |  1.0  ,  1.0  ,  1.0  |
```

```
C     FILT() = |  1.0  , -8.0  ,  1.0  |
C              |  1.0  ,  1.0  ,  1.0  |
C
      FILT08(CC) = -8.0
C
C --- 8-NEIGHBOR INVERSE DISTANCE WEIGHT MATRIX ---
C
C              |  1/DI**2  1/DX**2  1/DI**2  |
C     WEIT() = |  1/DY**2   .....   1/DY**2  |
C              |  1/DI**2  1/DX**2  1/DI**2  |
C
C      ..... = SUM NONZERO WEIT08() ELEMENTS THEN DIVIDE BY
C              ABSOLUTE VALUE OF FILT06(CC)
C
      DI = DSQRT( DX04*DX04 + DY04*DY04 )
      WEIT08(NW) = 1/DI**2
      WEIT08(WW) = 1/DY04**2
      WEIT08(SW) = 1/DI**2
      WEIT08(NN) = 1/DX04**2
      WEIT08(SS) = 1/DX04**2
      WEIT08(NE) = 1/DI**2
      WEIT08(EE) = 1/DY04**2
      WEIT08(SE) = 1/DI**2
      WEIT08(CC) = ( 4/DI**2 + 2/DX04**2 + 2/DY04**2 ) / 8
C
C --- BROADCAST PRODUCED MATRICES ---
C
 100  FORMAT(6F12.8)
 200  FORMAT(3F20.9)
 300  FORMAT(A)
C
      WRITE(2,300)' '
      WRITE(2,100)((INDX04(J,I),J=1,2),I=1,7,3)
      WRITE(2,100)((INDX04(J,I),J=1,2),I=2,8,3)
      WRITE(2,100)((INDX04(J,I),J=1,2),I=3,9,3)
      WRITE(2,300)' '
      WRITE(2,200)(FILT04(I),I=1,7,3)
      WRITE(2,200)(FILT04(I),I=2,8,3)
      WRITE(2,200)(FILT04(I),I=3,9,3)
      WRITE(2,300)' '
      WRITE(2,200)(WEIT04(I),I=1,7,3)
      WRITE(2,200)(WEIT04(I),I=2,8,3)
      WRITE(2,200)(WEIT04(I),I=3,9,3)
C
      WRITE(2,300)' '
      WRITE(2,100)((INDX06(J,I),J=1,2),I=1,7,3)
      WRITE(2,100)((INDX06(J,I),J=1,2),I=2,8,3)
      WRITE(2,100)((INDX06(J,I),J=1,2),I=3,9,3)
      WRITE(2,300)' '
      WRITE(2,200)(FILT06(I),I=1,7,3)
      WRITE(2,200)(FILT06(I),I=2,8,3)
```

```
      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)((INDX08(J,I),J=1,2),I=1,7,3)
      WRITE(2,100)((INDX08(J,I),J=1,2),I=2,8,3)
      WRITE(2,100)((INDX08(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)
      WRITE(2,300)' '
      WRITE(2,200)(WEIT08(I),I=1,7,3)
      WRITE(2,200)(WEIT08(I),I=2,8,3)
      WRITE(2,200)(WEIT08(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        (DELTA-X) (DELTA-Y)  =  (DELTA-X) (DELTA-Y)
C                4        4            6        6
C
C-----------------------------------------------------------------------
C ROGER BROWN , JUNE 1987
C-----------------------------------------------------------------------
C
      INCLUDE (GRIDDS)
C
      DX04 = 1.00
      DY04 = DX04
C
      DX06 = DX04*DY04 * 2/SQRT(3.0)
      DX06 = SQRT(DX06)
```

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

B.2.8 SURF.FOR

```
      SUBROUTINE SURF(XO,YO,ZO,LAP)
C
C-----------------------------------------------------------------------
C --- GRID POST VALUE & DERIVATIVE ----------------------------------
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
      INCLUDE (GRIDDS)
C
C --- POST ---
C
      R = DSQRT( XO*XO + YO*YO )
      U = R-RO
      ZO = 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 * XO/R
      ZDY = ZDU * YO/R
C
      LAP = -2*H/V * U * ZDU*ZDU
      LAP = LAP - ZDU/R
C
      RETURN
```

```
      END
```

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,Z6
```

```
            WRITE(2,'(3F20.9)')Z4,Z6,Z8
            WRITE(2,'(3F20.9)')ADD4,ADD6,ADD8
          ENDIF
C
  10  CONTINUE
C
      LAP4 = SUM04
      LAP6 = SUM06
      LAP8 = SUM08
C
      RETURN
      END
```

B.2.10 ELWIND.EXE

```
*
* EXECUTIVE ROUTINE FOR ELWIND SOFTWARE
*
FILEDEF 01 DISK ELWIND SAS
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
      INCLUDE(GRIDDS)
      REAL DIR
      CHARACTER*15 BLNK15/'    .          '/
```

```
      CHARACTER*20 BLNK20/'                    '/
C
      CALL CNVLTN
C
C --- 8-NEIGHBOR POSTS ---
C
      WRITE (1,'(A/A/A/A)')
     1' DATA WIND ;' ,
     2' INPUT' ,
     3' DIR8 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,BLNK15,       BLNK15,
     +                                  INDX06(X,DIR),INDX06(Y,DIR)
C        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-----------------------------------------------------------------------
C --- COMPARE DERIVATIVES VERSUS DIFFERENCES FOR RADIAL EDGE PROFILES --
C-----------------------------------------------------------------------
C
C     DEVELOP SURFACE OF ROTATION USING RADIAL EDGE PROFILE
C
C     EXAMINE RADIAL PATH AT VARIOUS ORIENTATIONS FROM CENTROID OF
C     CURVATURE
C
C-----------------------------------------------------------------------
C ROGER BROWN , SEPTEMBER 1987
C-----------------------------------------------------------------------
C
      INCLUDE (GRIDDS)
C
      REAL*8  ANGLE(0:6)
      REAL*8  DL4(0:6),DL6(0:6),DL8(0:6)
      INTEGER LUN,I
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(1.0D0)/90
      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
      RO=25.0
      H=1.0
      V=1.0
C
      WRITE(10,'(A/A/A/A/A)')
```

```
     1'/* GRID EDGE PROFILE COMPARISONS AT CRITICAL ANGLES */' ,
     2'DATA GRIDIF ;' ,
     3'INPUT ' ,
     4'RAD F45M90 S60M30 E90M45 ;' ,
     5'CARDS ;'
C
C --- COMPARE IDENTICAL EDGE PROFILE POINTS ON DIFFERENT RADIALS ---
C
      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,ZO,LAP)
            CALL FILTER( XO,YO, DL4(1),DL6(I),DL8(I) , 0,0 )
C           DL4(I) = DL4(I) - LAP
C           DL6(I) = DL6(I) - LAP
C           DL8(I) = DL8(I) - LAP
  20     CONTINUE
C
         DIF4 = DL4(3) - DL4(6)
         DIF6 = DL6(4) - DL6(2)
         DIF8 = DL8(6) - DL8(3)
C
         WRITE(10,'(F10.3,3F20.15)')J,DIF4,DIF6,DIF8
C
  25  CONTINUE
C
      WRITE(10,*)'SYMBOL1 C=BLACK I=SPLINE L=1 V=NONE ;'
      WRITE(10,*)'SYMBOL2 C=BLACK I=SPLINE L=2 V=NONE ;'
      WRITE(10,*)'SYMBOL3 C=BLACK I=SPLINE L=3 V=NONE ;'
      WRITE(10,*)'PROC GPLOT DATA=GRIDIF ;'
      WRITE(10,*)'PLOT F45M90*RAD S60M30*RAD E90M45*RAD / OVERLAY ;'
C
      STOP
      END
```

B.3 Data Files

B.3.1 EDGPRO.SAS

```
/* EDGE PROFILE */
DATA EDGPRO ;
INPUT
X Z DER1 DIF1 DER2 DIF2 ;
```

CARDS ;

0.00	-1.53081799	0.00159744	-0.00000099	0.00012759	0.00000974
0.25	-1.53041458	0.00162983	0.00000096	0.00013149	0.00000584
0.50	-1.53000259	0.00166320	0.00000001	0.00013555	-0.00001348
0.75	-1.52958298	0.00169761	-0.00000007	0.00013977	0.00001282
1.00	-1.52915382	0.00173310	0.00000068	0.00014417	-0.00000685
1.25	-1.52871609	0.00176972	0.00000030	0.00014876	0.00000382
1.50	-1.52826881	0.00180750	0.00000067	0.00015355	-0.00000096
1.75	-1.52781200	0.00184651	-0.00000020	0.00015855	-0.00000596
2.00	-1.52734566	0.00188679	-0.00000042	0.00016376	0.00000409
2.25	-1.52686882	0.00192841	-0.00000008	0.00016920	-0.00000136
2.50	-1.52638149	0.00197141	0.00000078	0.00017489	0.00000821
2.75	-1.52588272	0.00201587	0.00000019	0.00018084	-0.00001299
3.00	-1.52537346	0.00206186	-0.00000001	0.00018705	0.00001131
3.25	-1.52485180	0.00210943	0.00000201	0.00019356	0.00000480
3.50	-1.52431774	0.00215866	0.00000046	0.00020037	-0.00001727
3.75	-1.52377224	0.00220964	-0.00000093	0.00020751	0.00000612
4.00	-1.52321339	0.00226244	-0.00000033	0.00021498	-0.00000136
4.25	-1.52264118	0.00231716	0.00000027	0.00022282	0.00000606
4.50	-1.52205467	0.00237389	0.00000076	0.00023105	-0.00000217
4.75	-1.52145386	0.00243272	0.00000106	0.00023968	0.00000446
5.00	-1.52083778	0.00249377	0.00000105	0.00024875	-0.00000461
5.25	-1.52020645	0.00255714	0.00000062	0.00025829	0.00000111
5.50	-1.51955891	0.00262295	-0.00000035	0.00026831	-0.00000892
5.75	-1.51889515	0.00269134	-0.00000007	0.00027887	0.00001105
6.00	-1.51821327	0.00276243	0.00000132	0.00028998	-0.00000006
6.25	-1.51751328	0.00283638	-0.00000015	0.00030169	-0.00001177
6.50	-1.51679516	0.00291383	-0.00000081	0.00031404	0.00000640
6.75	-1.51605701	0.00299345	0.00000109	0.00032707	0.00000863
7.00	-1.51529789	0.00307692	0.00000154	0.00034083	-0.00000513
7.25	-1.51451778	0.00316393	0.00000036	0.00035537	-0.00000442
7.50	-1.51371574	0.00325468	0.00000117	0.00037075	0.00001072
7.75	-1.51288986	0.00334938	0.00000185	0.00038703	-0.00000556
8.00	-1.51204014	0.00344828	0.00000021	0.00040428	-0.00000755
8.25	-1.51116562	0.00355161	-0.00000013	0.00042257	0.00000468
8.50	-1.51026440	0.00365965	0.00000055	0.00044197	0.00000053
8.75	-1.50933552	0.00377270	0.00000195	0.00046258	0.00001044
9.00	-1.50837708	0.00389105	0.00000185	0.00048449	-0.00001147
9.25	-1.50738907	0.00401505	-0.00000009	0.00050780	-0.00000426
9.50	-1.50636959	0.00414507	-0.00000041	0.00053263	0.00000143
9.75	-1.50531673	0.00428151	0.00000049	0.00055910	0.00000547
10.00	-1.50422859	0.00442478	0.00000218	0.00058736	0.00000773
10.25	-1.50310326	0.00457535	0.00000229	0.00061755	-0.00000720
10.50	-1.50193977	0.00473373	0.00000222	0.00064984	0.00000629
10.75	-1.50073528	0.00490046	0.00000143	0.00068441	-0.00001303
11.00	-1.49948883	0.00507614	0.00000122	0.00072148	0.00001094
11.25	-1.49819660	0.00526142	0.00000095	0.00076127	-0.00001359
11.50	-1.49685764	0.00545703	0.00000181	0.00080404	0.00001994
11.75	-1.49546719	0.00566372	0.00000302	0.00085006	-0.00001082
12.00	-1.49402428	0.00588235	0.00000182	0.00089965	0.00000061
12.25	-1.49252510	0.00611387	0.00000109	0.00095317	-0.00000713

12.50	-1.49096680	0.00635930	0.00000171	0.00101102	0.00001132
12.75	-1.48934460	0.00661977	0.00000445	0.00107362	0.00000975
13.00	-1.48765469	0.00689655	0.00000424	0.00114150	-0.00001235
13.25	-1.48589420	0.00719101	0.00000160	0.00121520	-0.00000975
13.50	-1.48405838	0.00750469	0.00000264	0.00129537	0.00001689
13.75	-1.48214054	0.00783929	0.00000373	0.00138273	-0.00000943
14.00	-1.48013687	0.00819672	0.00000488	0.00147810	0.00001727
14.25	-1.47803974	0.00857909	0.00000589	0.00158242	-0.00001076
14.50	-1.47584438	0.00898876	0.00000439	0.00169675	-0.00000303
14.75	-1.47354317	0.00942840	0.00000535	0.00182234	0.00000871
15.00	-1.47112751	0.00990099	0.00000578	0.00196059	-0.00000747
15.25	-1.46858978	0.01040989	0.00000614	0.00211313	0.00000784
15.50	-1.46591949	0.01095890	0.00000835	0.00228185	0.00000697
15.75	-1.46310616	0.01155235	0.00000810	0.00246895	-0.00001228
16.00	-1.46013927	0.01219512	0.00000810	0.00267698	0.00000857
16.25	-1.45700455	0.01289283	0.00001039	0.00290894	0.00000549
16.50	-1.45368767	0.01365187	0.00001237	0.00316835	0.00000548
16.75	-1.45017242	0.01447964	0.00001240	0.00345939	-0.00001090
17.00	-1.44644165	0.01538461	0.00001341	0.00378698	0.00001246
17.25	-1.44247341	0.01637666	0.00001700	0.00415702	0.00000863
17.50	-1.43824482	0.01746725	0.00001933	0.00457657	0.00000107
17.75	-1.43373013	0.01866978	0.00002224	0.00505413	0.00001179
18.00	-1.42889881	0.02000000	0.00002525	0.00559999	-0.00000002
18.25	-1.42371750	0.02147651	0.00002694	0.00622674	-0.00000115
18.50	-1.41814709	0.02312139	0.00003192	0.00694978	0.00002349
18.75	-1.41214085	0.02496099	0.00003862	0.00778814	0.00000910
19.00	-1.40564728	0.02702703	0.00004397	0.00876552	0.00000829
19.25	-1.39860535	0.02935779	0.00005161	0.00991162	0.00002186
19.50	-1.39094257	0.03200000	0.00006063	0.01126399	0.00001225
19.75	-1.38257504	0.03501094	0.00007283	0.01287054	0.00003840
20.00	-1.37340069	0.03846154	0.00008789	0.01479290	0.00002339
20.25	-1.36330032	0.04244032	0.00010691	0.01711121	0.00005493
20.50	-1.35212708	0.04705882	0.00013089	0.01993079	0.00004296
20.75	-1.33970547	0.05245901	0.00015901	0.02339156	0.00006120
21.00	-1.32581806	0.05882353	0.00019938	0.02768166	0.00010460
21.25	-1.31019402	0.06639004	0.00025272	0.03305728	0.00011533
21.50	-1.29249668	0.07547164	0.00032067	0.03987176	0.00015205
21.75	-1.27229786	0.08648646	0.00041616	0.04861939	0.00023925
22.00	-1.24904537	0.09999996	0.00054592	0.05999994	0.00028753
22.25	-1.22202492	0.11678827	0.00072348	0.07501721	0.00042224
22.50	-1.19028950	0.13793099	0.00097549	0.09512478	0.00059360
22.75	-1.15257168	0.16494840	0.00133234	0.12243587	0.00083989
23.00	-1.10714912	0.19999999	0.00184327	0.15999997	0.00122440
23.25	-1.05165005	0.24615383	0.00255775	0.21207094	0.00164461
23.50	-0.98279375	0.30769229	0.00349736	0.28402358	0.00207585
23.75	-0.89605540	0.39024389	0.00454676	0.38072568	0.00197047
24.00	-0.78539819	0.50000000	0.00510883	0.50000000	-0.00016785
24.25	-0.64350110	0.63999999	0.00350082	0.61439991	-0.00710678
24.50	-0.46364760	0.79999995	-0.00295568	0.63999987	-0.01895285
24.75	-0.24497867	0.94117641	-0.01388121	0.44290650	-0.02195084
25.00	0.00000000	1.00000000	-0.02008533	0.00000000	0.00000000

25.25	0.24497867	0.94117641	-0.01388121	-0.44290650	0.02195084
25.50	0.46364760	0.79999995	-0.00295568	-0.63999987	0.01895285
25.75	0.64350110	0.63999999	0.00350082	-0.61439991	0.00710678
26.00	0.78539819	0.50000000	0.00510883	-0.50000000	0.00016785
26.25	0.89605540	0.39024389	0.00454676	-0.38072568	-0.00196570
26.50	0.98279375	0.30769229	0.00349736	-0.28402358	-0.00206345
26.75	1.05165005	0.24615383	0.00255775	-0.21207094	-0.00164366
27.00	1.10714912	0.19999999	0.00184327	-0.15999997	-0.00122440
27.25	1.15257168	0.16494840	0.00133234	-0.12243587	-0.00083989
27.50	1.19028950	0.13793099	0.00097549	-0.09512478	-0.00059360
27.75	1.22202492	0.11678827	0.00072348	-0.07501721	-0.00042224
28.00	1.24904537	0.09999996	0.00054592	-0.05999994	-0.00028753
28.25	1.27229786	0.08648646	0.00041616	-0.04861939	-0.00023925
28.50	1.29249668	0.07547164	0.00032067	-0.03987176	-0.00015205
28.75	1.31019402	0.06639004	0.00025272	-0.03305728	-0.00011533
29.00	1.32581806	0.05882353	0.00019938	-0.02768166	-0.00010460
29.25	1.33970547	0.05245901	0.00015901	-0.02339156	-0.00006120
29.50	1.35212708	0.04705882	0.00013089	-0.01993079	-0.00004296
29.75	1.36330032	0.04244852	0.00010691	-0.01711121	-0.00005493
30.00	1.37340069	0.03846154	0.00008789	-0.01479290	-0.00002339
30.25	1.38257504	0.03501094	0.00007283	-0.01287054	-0.00003840
30.50	1.39094257	0.03200000	0.00006063	-0.01126399	-0.00001225
30.75	1.39860535	0.02935779	0.00005161	-0.00991162	-0.00002186
31.00	1.40564728	0.02702703	0.00004397	-0.00876552	-0.00000829
31.25	1.41214085	0.02496099	0.00003862	-0.00778814	-0.00000910
31.50	1.41814709	0.02312139	0.00003192	-0.00694978	-0.00002349
31.75	1.42371750	0.02147651	0.00002694	-0.00622674	0.00000115
32.00	1.42889881	0.02000000	0.00002525	-0.00559999	0.00000002
32.25	1.43373013	0.01866978	0.00002224	-0.00505413	-0.00001179
32.50	1.43824482	0.01746725	0.00001933	-0.00457657	-0.00000107
32.75	1.44247341	0.01637666	0.00001700	-0.00415702	-0.00000863
33.00	1.44644165	0.01538461	0.00001341	-0.00378698	-0.00001246
33.25	1.45017242	0.01447964	0.00001240	-0.00345939	0.00001090
33.50	1.45368767	0.01365187	0.00001237	-0.00316835	-0.00000548
33.75	1.45700455	0.01289283	0.00001039	-0.00290894	-0.00000549
34.00	1.46013927	0.01219512	0.00000810	-0.00267698	-0.00000857
34.25	1.46310616	0.01155235	0.00000810	-0.00246895	0.00001228
34.50	1.46591949	0.01095890	0.00000835	-0.00228185	-0.00000697
34.75	1.46858978	0.01040989	0.00000614	-0.00211313	-0.00000784
35.00	1.47112751	0.00990099	0.00000578	-0.00196059	0.00000747
35.25	1.47354317	0.00942840	0.00000535	-0.00182234	-0.00000871
35.50	1.47584438	0.00898876	0.00000439	-0.00169675	0.00000303
35.75	1.47803974	0.00857909	0.00000589	-0.00158242	0.00001076
36.00	1.48013687	0.00819672	0.00000488	-0.00147810	-0.00001727
36.25	1.48214054	0.00783929	0.00000373	-0.00138273	0.00000943
36.50	1.48405838	0.00750469	0.00000264	-0.00129537	-0.00001689
36.75	1.48589420	0.00719101	0.00000160	-0.00121520	0.00000975
37.00	1.48765469	0.00689655	0.00000424	-0.00114150	0.00001235
37.25	1.48934460	0.00661977	0.00000445	-0.00107362	-0.00000975
37.50	1.49096680	0.00635930	0.00000171	-0.00101102	-0.00001132
37.75	1.49252510	0.00611387	0.00000109	-0.00095317	0.00000713

38.00	1.49402428	0.00588235	0.00000182	-0.00089965	-0.00000061
38.25	1.49546719	0.00566372	0.00000302	-0.00085006	0.00001082
38.50	1.49685764	0.00545703	0.00000181	-0.00080404	-0.00001994
38.75	1.49819660	0.00526142	0.00000095	-0.00076127	0.00001359
39.00	1.49948883	0.00507614	0.00000122	-0.00072148	-0.00001094
39.25	1.50073528	0.00490046	0.00000143	-0.00068441	0.00001303
39.50	1.50193977	0.00473373	0.00000222	-0.00064984	-0.00000629
39.75	1.50310326	0.00457535	0.00000229	-0.00061755	0.00000720
40.00	1.50422859	0.00442478	0.00000218	-0.00058736	-0.00000773
40.25	1.50531673	0.00428151	0.00000049	-0.00055910	-0.00000547
40.50	1.50636959	0.00414507	-0.00000041	-0.00053263	-0.00000143
40.75	1.50738907	0.00401505	-0.00000009	-0.00050780	0.00000426
41.00	1.50837708	0.00389105	0.00000185	-0.00048449	0.00001147
41.25	1.50933552	0.00377270	0.00000195	-0.00046258	-0.00001044
41.50	1.51026440	0.00365965	0.00000055	-0.00044197	-0.00000053
41.75	1.51116562	0.00355161	-0.00000013	-0.00042257	-0.00000468
42.00	1.51204014	0.00344828	0.00000021	-0.00040428	0.00000755
42.25	1.51288986	0.00334938	0.00000183	-0.00038703	0.00000556
42.50	1.51371574	0.00325468	0.00000117	-0.00037075	-0.00001072
42.75	1.51451778	0.00316393	0.00000036	-0.00035537	0.00000442
43.00	1.51529789	0.00307692	0.00000154	-0.00034083	0.00000513
43.25	1.51605701	0.00299345	0.00000109	-0.00032707	-0.00000863
43.50	1.51679516	0.00291333	-0.00000081	-0.00031404	-0.00000640
43.75	1.51751328	0.00283638	-0.00000015	-0.00030169	0.00001177
44.00	1.51821327	0.00276243	0.00000132	-0.00028998	0.00000006
44.25	1.51889515	0.00269134	-0.00000007	-0.00027887	-0.00001105
44.50	1.51955891	0.00262295	-0.00000035	-0.00026831	0.00000892
44.75	1.52020645	0.00255714	0.00000062	-0.00025829	-0.00000111
45.00	1.52083778	0.00249377	0.00000105	-0.00024875	0.00000461
45.25	1.52145386	0.00243272	0.00000106	-0.00023968	-0.00000446
45.50	1.52205467	0.00237389	0.00000076	-0.00023105	0.00000217
45.75	1.52264118	0.00231716	0.00000027	-0.00022282	-0.00000606
46.00	1.52321339	0.00226244	-0.00000033	-0.00021498	0.00000136
46.25	1.52377224	0.00220964	-0.00000093	-0.00020751	-0.00000612
46.50	1.52431774	0.00215866	0.00000046	-0.00020037	0.00001727
46.75	1.52485180	0.00210943	0.00000201	-0.00019356	-0.00000480
47.00	1.52537346	0.00206186	-0.00000001	-0.00018705	-0.00001131
47.25	1.52588272	0.00201587	0.00000019	-0.00018084	0.00001299
47.50	1.52638149	0.00197141	0.00000078	-0.00017489	-0.00000821
47.75	1.52686882	0.00192841	-0.00000008	-0.00016920	0.00000136
48.00	1.52734566	0.00188679	-0.00000042	-0.00016376	-0.00000409
48.25	1.52781200	0.00184651	-0.00000020	-0.00015855	0.00000596
48.50	1.52826881	0.00180750	0.00000067	-0.00015355	0.00000096
48.75	1.52871609	0.00176972	0.00000030	-0.00014876	-0.00000382
49.00	1.52915382	0.00173310	0.00000068	-0.00014417	0.00000685
49.25	1.52958298	0.00169761	-0.00000007	-0.00013977	-0.00001282
49.50	1.53000259	0.00166320	0.00000001	-0.00013555	0.00001348
49.75	1.53041458	0.00162983	0.00000096	-0.00013149	-0.00000584

```
SYMBOL1 C=BLACK I=SPLINE L=1 V=NONE ;
SYMBOL2 C=BLACK I=SPLINE L=2 V=NONE ;
SYMBOL3 C=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 P0 ;
INPUT
ANGO MLO 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    0.317444856
      3.00    0.618809854    0.145555718    0.132848772    0.317362931
      4.00    0.618809854    0.144798942    0.132593241    0.317247397
      5.00    0.618809854    0.143832334    0.132269669    0.317097451
      6.00    0.618809854    0.142660642    0.131881699    0.316912116
      7.00    0.618809854    0.141289624    0.131433692    0.316690276
      8.00    0.618809854    0.139726021    0.130930674    0.316430722
      9.00    0.618809854    0.137977521    0.130378277    0.316132208
     10.00    0.618809854    0.136052723    0.129782670    0.315793500
     11.00    0.618809854    0.133961098    0.129150489    0.315413443
     12.00    0.618809854    0.131712936    0.128488758    0.314991021
     13.00    0.618809854    0.129319303    0.127804807    0.314525421
     14.00    0.618809854    0.126791978    0.127106189    0.314016094
     15.00    0.618809854    0.124143399    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    0.112604739    0.123662317    0.310820395
     20.00    0.618809854    0.109554929    0.123036074    0.310059234
     21.00    0.618809854    0.106468591    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    0.088032694    0.120015929    0.303954955
     28.00    0.618809854    0.085113888    0.119834988    0.303030056
     29.00    0.618809854    0.082277788    0.119725650    0.302109662
     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    0.119834990    0.299442587
     33.00    0.618809854    0.072027112    0.120015931    0.298607625
     34.00    0.618809854    0.069799576    0.120266552    0.297810912
```

35.00	0.618809854	0.067728470	0.120584187	0.297058928
36.00	0.618809854	0.065823687	0.120965453	0.296357870
37.00	0.618809854	0.064094306	0.121406282	0.295713583
38.00	0.618809854	0.062548557	0.121901961	0.295131485
39.00	0.618809854	0.061193782	0.122447180	0.294616508
40.00	0.618809854	0.060036405	0.123036082	0.294173041
41.00	0.618809854	0.059081907	0.123662326	0.293804880
42.00	0.618809854	0.058334803	0.124319146	0.293515189
43.00	0.618809854	0.057798624	0.124999428	0.293306462
44.00	0.618809854	0.057475901	0.125695779	0.293180500
45.00	0.618809854	0.057368157	0.126400606	0.293138392
46.00	0.618809854	0.057475901	0.127106200	0.293180500
47.00	0.618809854	0.057798624	0.127804819	0.293306462
48.00	0.618809854	0.058334803	0.128488771	0.293515189
49.00	0.618809854	0.059081907	0.129150502	0.293804880
50.00	0.618809854	0.060036405	0.129782683	0.294173041
51.00	0.618809854	0.061193782	0.130378290	0.294616508
52.00	0.618809854	0.062548557	0.130930688	0.295131485
53.00	0.618809854	0.064094306	0.131433706	0.295713583
54.00	0.618809854	0.065823687	0.131881714	0.296357870
55.00	0.618809854	0.067728470	0.132269684	0.297058928
56.00	0.618809854	0.069799576	0.132593256	0.297810912
57.00	0.618809854	0.072027112	0.132848788	0.298607625
58.00	0.618809854	0.074400415	0.133033399	0.299442587
59.00	0.618809854	0.076908102	0.133145005	0.300309119
60.00	0.618809854	0.079538116	0.133182348	0.301200422
61.00	0.618809854	0.082277788	0.133145006	0.302109662
62.00	0.618809854	0.085113888	0.133033399	0.303030056
63.00	0.618809854	0.088032694	0.132848788	0.303954955
64.00	0.618809854	0.091020054	0.132593257	0.304877923
65.00	0.618809854	0.094061455	0.132269685	0.305792813
66.00	0.618809854	0.097142094	0.131881714	0.306693838
67.00	0.618809854	0.100246949	0.131433707	0.307575630
68.00	0.618809854	0.103360858	0.130930689	0.308433291
69.00	0.618809854	0.106468591	0.130378292	0.309262439
70.00	0.618809854	0.109554929	0.129782685	0.310059234
71.00	0.618809854	0.112604739	0.129150504	0.310820395
72.00	0.618809854	0.115603053	0.128488772	0.311543213
73.00	0.618809854	0.118535144	0.127804821	0.312225538
74.00	0.618809854	0.121386601	0.127106203	0.312865769
75.00	0.618809854	0.124143399	0.126400608	0.313462822
76.00	0.618809854	0.126791978	0.125695782	0.314016094
77.00	0.618809854	0.129319303	0.124999431	0.314525421
78.00	0.618809854	0.131712936	0.124319150	0.314991021
79.00	0.618809854	0.133961098	0.123662329	0.315413443
80.00	0.618809854	0.136052723	0.123036086	0.315793500
81.00	0.618809854	0.137977521	0.122447184	0.316132208
82.00	0.618809854	0.139726021	0.121901966	0.316430722
83.00	0.618809854	0.141289624	0.121406287	0.316690276
84.00	0.618809854	0.142660642	0.120965458	0.316912116
85.00	0.618809854	0.143832334	0.120584193	0.317097451

```
      86.00    0.618809854    0.144798942    0.120266558    0.317247397
      87.00    0.618809854    0.145555718    0.120015938    0.317362931
      88.00    0.618809854    0.146098942    0.119834997    0.317444856
      89.00    0.618809854    0.146425947    0.119725658    0.317493766
      90.00    0.618809854    0.146535128    0.119689082    0.317510028
DATA P1 ;
INPUT
ANG1 ML1 L41 L61 L81 ;
CARDS ;
       0.00   -0.040000000    0.079978687    0.086578865    0.100020038
       1.00   -0.040000000    0.079960498    0.086589878    0.100020091
       2.00   -0.040000000    0.079906052    0.086622805    0.100020348
       3.00   -0.040000000    0.079815713    0.086677314    0.100021104
       4.00   -0.040000000    0.079690080    0.086752853    0.100022842
       5.00   -0.040000000    0.079529985    0.086848655    0.100026225
       6.00   -0.040000000    0.079336480    0.086963746    0.100032083
       7.00   -0.040000000    0.079110827    0.087096948    0.100041395
       8.00   -0.040000000    0.078854485    0.087246893    0.100055271
       9.00   -0.040000000    0.078569094    0.087412030    0.100074927
      10.00   -0.040000000    0.078256458    0.087590641    0.100101660
      11.00   -0.040000000    0.077918525    0.087780857    0.100136822
      12.00   -0.040000000    0.077557373    0.087980669    0.100181788
      13.00   -0.040000000    0.077175186    0.088187952    0.100237926
      14.00   -0.040000000    0.076774236    0.088400486    0.100306566
      15.00   -0.040000000    0.076356865    0.088615973    0.100388966
      16.00   -0.040000000    0.075925462    0.088832064    0.100486279
      17.00   -0.040000000    0.075482449    0.089046384    0.100599523
      18.00   -0.040000000    0.075030260    0.089256558    0.100729544
      19.00   -0.040000000    0.074571326    0.089460238    0.100876993
      20.00   -0.040000000    0.074108058    0.089655131    0.101042291
      21.00   -0.040000000    0.073642834    0.089839024    0.101225605
      22.00   -0.040000000    0.073177986    0.090009816    0.101426827
      23.00   -0.040000000    0.072715789    0.090165542    0.101645552
      24.00   -0.040000000    0.072258448    0.090304400    0.101881061
      25.00   -0.040000000    0.071808093    0.090424773    0.102132313
      26.00   -0.040000000    0.071366770    0.090525254    0.102397937
      27.00   -0.040000000    0.070936432    0.090604662    0.102676233
      28.00   -0.040000000    0.070518941    0.090662062    0.102965182
      29.00   -0.040000000    0.070116056    0.090696776    0.103262451
      30.00   -0.040000000    0.069729436    0.090708393    0.103565423
      31.00   -0.040000000    0.069360634    0.090696776    0.103871220
      32.00   -0.040000000    0.069011102    0.090662062    0.104176741
      33.00   -0.040000000    0.068682180    0.090604662    0.104478703
      34.00   -0.040000000    0.068375108    0.090525253    0.104773693
      35.00   -0.040000000    0.068091016    0.090424772    0.105058223
      36.00   -0.040000000    0.067830934    0.090304399    0.105328793
      37.00   -0.040000000    0.067595784    0.090165541    0.105581954
      38.00   -0.040000000    0.067386390    0.090009815    0.105814379
      39.00   -0.040000000    0.067203472    0.089839023    0.106022928
      40.00   -0.040000000    0.067047652    0.089655129    0.106204716
      41.00   -0.040000000    0.066919454    0.089460237    0.106357182
```

42.00	-0.040000000	0.066819304	0.089256557	0.106478138
43.00	-0.040000000	0.066747533	0.089046382	0.106565827
44.00	-0.040000000	0.066704376	0.088832062	0.106618964
45.00	-0.040000000	0.066689975	0.088615971	0.106636764
46.00	-0.040000000	0.066704376	0.088400484	0.106618964
47.00	-0.040000000	0.066747533	0.088187950	0.106565827
48.00	-0.040000000	0.066819304	0.087980666	0.106478138
49.00	-0.040000000	0.066919454	0.087780854	0.106357182
50.00	-0.040000000	0.067047652	0.087590639	0.106204716
51.00	-0.040000000	0.067203472	0.087412027	0.106022928
52.00	-0.040000000	0.067386390	0.087246890	0.105814379
53.00	-0.040000000	0.067595784	0.087096946	0.105581954
54.00	-0.040000000	0.067830934	0.086963744	0.105328793
55.00	-0.040000000	0.068091016	0.086848653	0.105058223
56.00	-0.040000000	0.068375108	0.086752850	0.104773693
57.00	-0.040000000	0.068682180	0.086677311	0.104478703
58.00	-0.040000000	0.069011102	0.086622802	0.104176741
59.00	-0.040000000	0.069360634	0.086589875	0.103871220
60.00	-0.040000000	0.069729436	0.086578862	0.103565423
61.00	-0.040000000	0.070116056	0.086589875	0.103262451
62.00	-0.040000000	0.070518941	0.086622802	0.102965182
63.00	-0.040000000	0.070936432	0.086677311	0.102676233
64.00	-0.040000000	0.071366770	0.086752850	0.102397937
65.00	-0.040000000	0.071808093	0.086848653	0.102132313
66.00	-0.040000000	0.072258448	0.086963743	0.101881061
67.00	-0.040000000	0.072715789	0.087096946	0.101645552
68.00	-0.040000000	0.073177986	0.087246890	0.101426827
69.00	-0.040000000	0.073642834	0.087412027	0.101225605
70.00	-0.040000000	0.074108058	0.087590639	0.101042291
71.00	-0.040000000	0.074571326	0.087780854	0.100876993
72.00	-0.040000000	0.075030260	0.087980666	0.100729544
73.00	-0.040000000	0.075482449	0.088187950	0.100599523
74.00	-0.040000000	0.075925462	0.088400484	0.100486279
75.00	-0.040000000	0.076356865	0.088615970	0.100388966
76.00	-0.040000000	0.076774236	0.088832061	0.100306566
77.00	-0.040000000	0.077175186	0.089046381	0.100237926
78.00	-0.040000000	0.077557373	0.089256556	0.100181788
79.00	-0.040000000	0.077918525	0.089460236	0.100136822
80.00	-0.040000000	0.078256458	0.089655128	0.100101660
81.00	-0.040000000	0.078569094	0.089839022	0.100074927
82.00	-0.040000000	0.078854485	0.090009814	0.100055271
83.00	-0.040000000	0.079110827	0.090165540	0.100041395
84.00	-0.040000000	0.079336480	0.090304398	0.100032083
85.00	-0.040000000	0.079529985	0.090424771	0.100026225
86.00	-0.040000000	0.079690080	0.090525252	0.100022842
87.00	-0.040000000	0.079815713	0.090604660	0.100021104
88.00	-0.040000000	0.079906052	0.090662060	0.100020348
89.00	-0.040000000	0.079960498	0.090696774	0.100020091
90.00	-0.040000000	0.079978687	0.090708392	0.100020038

```
DATA P2 ;
INPUT
```

```
ANG2 ML2 L42 L62 L82 ;
CARDS ;
```

0.00	-0.678841872	0.266599086	0.013550035	0.151982701
1.00	-0.678841872	0.266485442	0.013579943	0.151946194
2.00	-0.678841872	0.266145032	0.013669322	0.151836750
3.00	-0.678841872	0.265579427	0.013817147	0.151654609
4.00	-0.678841872	0.264791237	0.014021720	0.151400179
5.00	-0.678841872	0.263784103	0.014280695	0.151074052
6.00	-0.678841872	0.262562687	0.014591111	0.150677015
7.00	-0.678841872	0.261132650	0.014949426	0.150210080
8.00	-0.678841872	0.259500635	0.015351562	0.149674501
9.00	-0.678841872	0.257674242	0.015792962	0.149071805
10.00	-0.678841872	0.255661999	0.016268639	0.148403813
11.00	-0.678841872	0.253473330	0.016773242	0.147672673
12.00	-0.678841872	0.251118515	0.017301121	0.146880879
13.00	-0.678841872	0.248608652	0.017846390	0.146031291
14.00	-0.678841872	0.245955610	0.018403001	0.145127158
15.00	-0.678841872	0.243171979	0.018964812	0.144172118
16.00	-0.678841872	0.240271016	0.019525651	0.143170211
17.00	-0.678841872	0.237266586	0.020079393	0.142125871
18.00	-0.678841872	0.234175102	0.020620020	0.141043914
19.00	-0.678841872	0.231005453	0.021141686	0.139929518
20.00	-0.678841872	0.227778942	0.021638777	0.138788196
21.00	-0.678841872	0.224509208	0.022105972	0.137625754
22.00	-0.678841872	0.221212149	0.022538289	0.136448255
23.00	-0.678841872	0.217903848	0.022931140	0.135261959
24.00	-0.678841872	0.214600692	0.023280372	0.134073278
25.00	-0.678841872	0.211318285	0.023582306	0.132888705
26.00	-0.678841872	0.208073372	0.023833769	0.131714763
27.00	-0.678841872	0.204881751	0.024032128	0.130557937
28.00	-0.678841872	0.201759194	0.024175309	0.129424615
29.00	-0.678841872	0.198721160	0.024261820	0.128321030
30.00	-0.678841872	0.195782717	0.024290757	0.127253208
31.00	-0.678841872	0.192958464	0.024261819	0.126226915
32.00	-0.678841872	0.190262450	0.024175308	0.125247622
33.00	-0.678841872	0.187708103	0.024032125	0.124320464
34.00	-0.678841872	0.185308158	0.023833765	0.123450215
35.00	-0.678841872	0.183074593	0.023582301	0.122641267
36.00	-0.678841872	0.181018562	0.023280367	0.121897616
37.00	-0.678841872	0.179150339	0.022931134	0.121222854
38.00	-0.678841872	0.177479263	0.022538282	0.120620168
39.00	-0.678841872	0.176013692	0.022105963	0.120092342
40.00	-0.678841872	0.174760959	0.021638768	0.119641764
41.00	-0.678841872	0.173727333	0.021141676	0.119270430
42.00	-0.678841872	0.172917985	0.020620009	0.118979959
43.00	-0.678841872	0.172336969	0.020079382	0.118771597
44.00	-0.678841872	0.171987191	0.019525639	0.118646228
45.00	-0.678841872	0.171870404	0.018964799	0.118604380
46.00	-0.678841872	0.171987191	0.018402988	0.118646228
47.00	-0.678841872	0.172336969	0.017846376	0.118771597
48.00	-0.678841872	0.172917985	0.017301106	0.118979959

```
      49.00   -0.678841872   0.173727333   0.016773227   0.119270430
      50.00   -0.678841872   0.174760959   0.016268623   0.119641764
      51.00   -0.678841872   0.176013692   0.015792945   0.120092342
      52.00   -0.678841872   0.177479263   0.015351545   0.120620168
      53.00   -0.678841872   0.179150339   0.014949408   0.121222854
      54.00   -0.678841872   0.181018562   0.014591093   0.121897616
      55.00   -0.678841872   0.183074593   0.014280677   0.122641267
      56.00   -0.678841872   0.185308158   0.014021701   0.123450215
      57.00   -0.678841872   0.187708103   0.013817128   0.124320464
      58.00   -0.678841872   0.190262450   0.013669303   0.125247622
      59.00   -0.678841872   0.192958464   0.013579923   0.126226915
      60.00   -0.678841872   0.195782717   0.013550015   0.127253208
      61.00   -0.678841872   0.198721160   0.013579923   0.128321030
      62.00   -0.678841872   0.201759194   0.013669303   0.129424615
      63.00   -0.678841872   0.204881751   0.013817127   0.130557937
      64.00   -0.678841872   0.208073572   0.014021700   0.131714763
      65.00   -0.678841872   0.211318285   0.014280675   0.132888705
      66.00   -0.678841872   0.214600492   0.014591091   0.134073278
      67.00   -0.678841872   0.217903868   0.014949406   0.135261959
      68.00   -0.678841872   0.221212149   0.015351543   0.136448255
      69.00   -0.678841872   0.224509208   0.015792943   0.137625754
      70.00   -0.678841872   0.227778942   0.016268620   0.138788196
      71.00   -0.678841872   0.231005453   0.016773223   0.139929518
      72.00   -0.678841872   0.234173102   0.017301102   0.141043914
      73.00   -0.678841872   0.237266586   0.017846372   0.142125871
      74.00   -0.678841872   0.240271016   0.018402984   0.143170211
      75.00   -0.678841872   0.243171979   0.018964794   0.144172118
      76.00   -0.678841872   0.245955610   0.019525634   0.145127158
      77.00   -0.678841872   0.248608652   0.020079377   0.146031291
      78.00   -0.678841872   0.251118515   0.020620004   0.146880879
      79.00   -0.678841872   0.253473330   0.021141670   0.147672673
      80.00   -0.678841872   0.255661999   0.021638762   0.148403813
      81.00   -0.678841872   0.257674242   0.022105957   0.149071805
      82.00   -0.678841872   0.259500635   0.022538274   0.149674501
      83.00   -0.678841872   0.261132650   0.022931126   0.150210080
      84.00   -0.678841872   0.262562687   0.023280359   0.150677015
      85.00   -0.678841872   0.263784103   0.023582292   0.151074052
      86.00   -0.678841872   0.264791257   0.023833756   0.151400179
      87.00   -0.678841872   0.265579427   0.024032115   0.151654609
      88.00   -0.678841872   0.266145032   0.024175298   0.151836750
      89.00   -0.678841872   0.266485442   0.024261808   0.151946194
      90.00   -0.678841872   0.266599086   0.024290746   0.151982701
DATA P3 ;
   SET P0;
   SET P1;
   SET P2;
SYMBOL1 C=BLACK I=SPLINE L=1 V=NONE ;
SYMBOL2 C=BLACK I=SPLINE L=2 V=NONE ;
SYMBOL3 C=BLACK I=SPLINE L=3 V=NONE ;
SYMBOL4 C=BLACK I=SPLINE L=1 V=NONE ;
SYMBOL5 C=BLACK I=SPLINE 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
DIR8 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 ;
```

B.3.4 RADIAL.SAS

```
/* GRID EDGE PROFILE COMPARISONS AT CRITICAL ANGLES */
DATA GRIDIF ;
INPUT
RAD F45M90 S60M30 E90M45 ;
CARDS ;
    12.500  -0.000000384110237  -0.000000002015516  -0.000000410492073
    12.625  -0.000000461531525  -0.000000001728468  -0.000000490058917
    12.750  -0.000000543324767  -0.000000001436148  -0.000000574218154
    12.875  -0.000000629931741  -0.000000001136224  -0.000000663433401
    13.000  -0.000000721830874  -0.000000000826200  -0.000000758207472
    13.125  -0.000000819541243  -0.000000000503381  -0.000000859086669
    13.250  -0.000000923627003  -0.000000000164837  -0.000000966665560
    13.375  -0.000001034702288   0.000000000192630  -0.000001081592289
    13.500  -0.000001153436675   0.000000000572545  -0.000001204574473
    13.625  -0.000001280561247   0.000000000978795  -0.000001336385787
    13.750  -0.000001416875359   0.000000001415701  -0.000001477873307
    13.875  -0.000001563254187   0.000000001888073  -0.000001629965724
    14.000  -0.000001720657153   0.000000002401291  -0.000001793682545
    14.125  -0.000001890137373   0.000000002961391  -0.000001970144390
    14.250  -0.000002072852225   0.000000003575162  -0.000002160584570
    14.375  -0.000002270075213   0.000000004250260  -0.000002366362094
    14.500  -0.000002483209298   0.000000004995338  -0.000002588976312
    14.625  -0.000002713801890   0.000000005820201  -0.000002830083415
    14.750  -0.000002963561738   0.000000006735976  -0.000003091515076
    14.875  -0.000003234377982   0.000000007755319  -0.000003375299502
    15.000  -0.000003528341679   0.000000008892647  -0.000003683685277
    15.125  -0.000003847770131   0.000000010164414  -0.000004019168387
    15.250  -0.000004195234466   0.000000011589429  -0.000004384522902
    15.375  -0.000004573590902   0.000000013189230  -0.000004782835852
    15.500  -0.000004986016271   0.000000014988515  -0.000005217546957
    15.625  -0.000005436048428   0.000000017015656  -0.000005692493917
    15.750  -0.000005927632283   0.000000019303294  -0.000006211964158
    15.875  -0.000006465172337   0.000000021889042  -0.000006780754028
    16.000  -0.000007053592708   0.000000024816312  -0.000007404236642
    16.125  -0.000007698405852   0.000000028135288  -0.000008088439747
    16.250  -0.000008405791342   0.000000031904081  -0.000008840135280
    16.375  -0.000009182686349   0.000000036190089  -0.000009666942507
    16.500  -0.000010036889726   0.000000041071614  -0.000010577447076
    16.625  -0.000010977181962   0.000000046639785  -0.000011581338630
    16.750  -0.000012013463652   0.000000053000845  -0.000012689570237
    16.875  -0.000013156915665   0.000000060278873  -0.000013914543407
    17.000  -0.000014420184729   0.000000068619060  -0.000015270323255
    17.125  -0.000015817598896   0.000000078191599  -0.000016772889227
    17.250  -0.000017365418187   0.000000089196388  -0.000018440427878
    17.375  -0.000019082126731   0.000000101868675  -0.000020293675490
    17.500  -0.000020988774007   0.000000116485858  -0.000022356319882
    17.625  -0.000023109374244   0.000000133375716  -0.000024655472717
```

17.750	-0.000025471374927	0.000000152926376	-0.000027222225931
17.875	-0.000028106207569	0.000000175598417	-0.000030092308837
18.000	-0.000031049936674	0.000000201939594	-0.000033306865963
18.125	-0.000034344026157	0.000000232602822	-0.000036913380086
18.250	-0.000038036246632	0.000000268368154	-0.000040966770334
18.375	-0.000042181752065	0.000000310169741	-0.000045530701896
18.500	-0.000046844360562	0.000000359128968	-0.000050679152234
18.625	-0.000052098081855	0.000000416595287	-0.000056498288991
18.750	-0.000058028943696	0.000000484196672	-0.000063088727769
18.875	-0.000064737181381	0.000000563902081	-0.000070568254013
19.000	-0.000072339869593	0.000000658099036	-0.000079075113581
19.125	-0.000080974094509	0.000000769690142	-0.000088772001836
19.250	-0.000090800787522	0.000000902213494	-0.000099850913144
19.375	-0.000102009371398	0.000001059993212	-0.000112539052754
19.500	-0.000114823406682	0.000001248327993	-0.000127106063613
19.625	-0.000129507472750	0.000001473727758	-0.000143872884348
19.750	-0.000146375576644	0.000001744211096	-0.000163222634616
19.875	-0.000165801456834	0.000002069679564	-0.000185614024399
20.000	-0.000188231242381	0.000002462388994	-0.000211597908961
20.125	-0.000214199045429	0.000002937542928	-0.000241837766394
20.250	-0.000244346212360	0.000003514039077	-0.000277135064637
20.375	-0.000279445143181	0.000004215406117	-0.000318460713595
20.500	-0.000320428817110	0.000005070974580	-0.000366994065444
20.625	-0.000368427442345	0.000006117330669	-0.000424171224577
20.750	-0.000424813985090	0.000007400102673	-0.000491744733451
20.875	-0.000491260727272	0.000008976120856	-0.000571856954888
21.000	-0.000569809442096	0.000010915962792	-0.000667129558190
21.125	-0.000662958224836	0.000013306828190	-0.000780771207051
21.250	-0.000773768388485	0.000016255545753	-0.000916704411496
21.375	-0.000905994934782	0.000019891237284	-0.001079709752227
21.500	-0.001064243799973	0.000024366640220	-0.001275579880653
21.625	-0.001254156809274	0.000029856122166	-0.001511264285620
21.750	-0.001482622272680	0.000036546663308	-0.001794964304627
21.875	-0.001757999077725	0.000044614931516	-0.002136098486274
22.000	-0.002090326199998	0.000054177995691	-0.002544988130832
22.125	-0.002491456826590	0.000065195483687	-0.003031990560202
22.250	-0.002974997505723	0.000077284312352	-0.003605601627720
22.375	-0.003555827249867	0.000089379361537	-0.004268717547568
22.500	-0.004248787284049	0.000099129390647	-0.005011752297114
22.625	-0.005065819591685	0.000101853188131	-0.005800672028837
22.750	-0.006010323892551	0.000088802678041	-0.006557459158219
22.875	-0.007066729144845	0.000044429237708	-0.007130813710714
23.000	-0.008182256868195	-0.000057534967433	-0.007257865662562
23.125	-0.009236660681594	-0.000258731367264	-0.006526613078333
23.250	-0.009996271234626	-0.000619218017846	-0.004367260215785
23.375	-0.010051151505999	-0.001212300328128	-0.000126331500393
23.500	-0.008747795345260	-0.002097546485463	0.006710074503194
23.625	-0.005155837019877	-0.003252672609728	0.016123551998978
23.750	0.001850575202989	-0.004455413183600	0.027123600984198
23.875	0.013286208822686	-0.005157578133656	0.037570076744855
24.000	0.029460469549676	-0.004485766994050	0.044606683851822

24.125	0.049225261182722	-0.001549861721109	0.045754463929391
24.250	0.069471067162111	0.003913408657621	0.040173041337456
24.375	0.085378917938251	0.010894414238693	0.029284134095190
24.500	0.091627618123979	0.017037706959322	0.016270042655378
24.625	0.084185459432757	0.019474408545311	0.004699926818758
24.750	0.061939818423043	0.016251227620973	-0.002954682334514
24.875	0.027481836915307	0.007541363798939	-0.006217352155226
25.000	-0.013288712059669	-0.004129530887772	-0.006616726495212
25.125	-0.052656542010486	-0.014857830925478	-0.006889876294491
25.250	-0.083204808636154	-0.021068578280603	-0.009719558369515
25.375	-0.099827758554692	-0.021071086120300	-0.016491684429980
25.500	-0.100974528864704	-0.015664282906164	-0.026561538226175
25.625	-0.088760092694334	-0.007520440020235	-0.037365137340504
25.750	-0.067954788104046	0.000245795847746	-0.045421967378023
25.875	-0.044240283778566	0.005476222138697	-0.047858925295431
26.000	-0.022433020230375	0.007590158016059	-0.043702978794106
26.125	-0.005393668436264	0.007274297610651	-0.034211712945623
26.250	0.006075641580575	0.005696867322695	-0.022098726257664
26.375	0.012611676411602	0.003845355068370	-0.010231370472881
26.500	0.015481262222027	0.002269603844159	-0.000601654451298
26.625	0.015977934876519	0.001143070036548	0.006007379461521
26.750	0.015130421177753	0.000430540555310	0.009769021433226
26.875	0.013642349270635	0.000025042404291	0.011340041974058
27.000	0.011943803667143	-0.000179809947944	0.011465375188525
27.125	0.010271784669936	-0.000264885217763	0.010765818038809
27.250	0.008741180111419	-0.000283938869391	0.009677201308010
27.375	0.007395818630821	-0.000269800525754	0.008469910880517
27.500	0.006241313046385	-0.000241274601631	0.007293920250585
27.625	0.005264670081901	-0.000208502906461	0.006221707853962
27.750	0.004445317323611	-0.000176568051728	0.005280207598989
27.875	0.003760938962332	-0.000147730212934	0.004471752388012
28.000	0.003190325834183	-0.000122746845368	0.003786715909022
28.125	0.002714582031463	-0.000101622465891	0.003210664835412
28.250	0.002317469486429	-0.000084020451917	0.002728153432656
28.375	0.001985329818707	-0.000069480842201	0.002324574406567
28.500	0.001706822292223	-0.000057529698860	0.001986936044951
28.625	0.001472602676728	-0.000047729421768	0.001704075478122
28.750	0.001275004533093	-0.000039697912034	0.001466595762704
28.875	0.001107750256724	-0.000033112036669	0.001266683517924
29.000	0.000965701363589	-0.000027703773855	0.001097889038624
29.125	0.000844648651787	-0.000023253459559	0.000954909219212
29.250	0.000741138868635	-0.000019582373580	0.000833391132819
29.375	0.000652333045349	-0.000016545720973	0.000729762367524
29.500	0.000575891458796	-0.000014026440205	0.000641088405668
29.625	0.000509880557914	-0.000011929951584	0.000564954711118
29.750	0.000452697792924	-0.000010179807840	0.000499370237820
29.875	0.000403010928069	-0.000008714145251	0.000442688953744
30.000	0.000359709019461	-0.000007482815257	0.000393546236249
30.125	0.000321862765639	-0.000006445079647	0.000350807401490
30.250	0.000288692381926	-0.000005567764978	0.000313526063328
30.375	0.000259541515053	-0.000004823787304	0.000280910421136

30.500	0.000233856011407	-0.000004190973510	0.000252295929127
30.625	0.000211166590891	-0.000003651119264	0.000227123097904
30.750	0.000191074669507	-0.000003189235326	0.000204919424334
30.875	0.000173240725803	-0.000002792943612	0.000185284645659
31.000	0.000157374727502	-0.000002451992362	0.000167878674368
31.125	0.000143228230746	-0.000002157866077	0.000152411699143
31.250	0.000130587840940	-0.000001903470979	0.000138636039884
31.375	0.000119269784982	-0.000001682880717	0.000126339426656
31.500	0.000109115393161	-0.000001491130271	0.000115339437499
31.625	0.000099987327680	-0.000001324048464	0.000105478881955
31.750	0.000091766425699	-0.000001178121480	0.000096621958437
31.875	0.000084349049535	-0.000001050381356	0.000088651046618
32.000	0.000077644856607	-0.000000938314621	0.000081464022345
32.125	0.000071574917686	-0.000000839787246	0.000074972003724
32.250	0.000066070124983	-0.000000752962822	0.000069097453982
32.375	0.000061069842005	-0.000000676351503	0.000063772580379
32.500	0.000056520755672	-0.000000608547722	0.000058937979420
32.625	0.000052375898022	-0.000000548495089	0.000054541487544
32.750	0.000048593810501	-0.000000495157171	0.000050537203688
32.875	0.000045137828400	-0.000000447713110	0.000046884655989
33.000	0.000041975466801	-0.000000405437227	0.000043548089667
33.125	0.000039077892476	-0.000000367701916	0.000040495857045
33.250	0.000036419468722	-0.000000333963271	0.000037699893879
33.375	0.000033977362269	-0.000000303748963	0.000035135268759
33.500	0.000031731203061	-0.000000276648028	0.000032779794585
33.625	0.000029662789254	-0.000000252302192	0.000030613692818
33.750	0.000027755830898	-0.000000230398540	0.000028619302756
33.875	0.000025995726814	-0.000000210663270	0.000026780829306
34.000	0.000024369370011	-0.000000192856375	0.000025084123694
34.125	0.000022864977692	-0.000000176767106	0.000023516492453
34.250	0.000021471942449	-0.000000162210091	0.000022066530769
34.375	0.000020180701829	-0.000000149022006	0.000020723976755
34.500	0.000018982623768	-0.000000137068720	0.000019479583840
34.625	0.000017869905816	-0.000000126192840	0.000018325008820
34.750	0.000016835486359	-0.000000116311588	0.000017252713458
34.875	0.000015872966261	-0.000000107314975	0.000016255877888
35.000	0.000014976539626	-0.000000099114224	0.000015328324251
35.125	0.000014140932492	-0.000000091630392	0.000014464449279
35.250	0.000013361348526	-0.000000084793186	0.000013659164634
35.375	0.000012633420770	-0.000000078539925	0.000012907844108
35.500	0.000011953168787	-0.000000072814644	0.000012206276745
35.625	0.000011316960491	-0.000000067567306	0.000011550625231
35.750	0.000010721478128	-0.000000062753116	0.000010937388852
35.875	0.000010163687919	-0.000000058331924	0.000010363370510
36.000	0.000009640812925	-0.000000054267692	0.000009825647299
36.125	0.000009150308773	-0.000000050528042	0.000009321544220
36.250	0.000008689841897	-0.000000047083845	0.000008848610677
36.375	0.000008257270039	-0.000000043908865	0.000008404599428
36.500	0.000007850624716	-0.000000040979444	0.000007987447713
36.625	0.000007468095479	-0.000000038274228	0.000007595260319
36.750	0.000007108015724	-0.000000035773918	0.000007226294352

```
   36.875   0.000006768849927  -0.000000033461056   0.000006878945536
   37.000   0.000006449182107  -0.000000031319831   0.000006551735883
   37.125   0.000006147705427  -0.000000029335911   0.000006243302560
   37.250   0.000005863212781  -0.000000027496289   0.000005952387848
   37.375   0.000005594588280  -0.000000025789152   0.000005677830068
   37.500   0.000005340799545  -0.000000024203757   0.000005418555360
SYMBOL1 C=BLACK I=SPLINE L=1 V=NONE ;
SYMBOL2 C=BLACK I=SPLINE L=2 V=NONE ;
SYMBOL3 C=BLACK I=SPLINE L=3 V=NONE ;
PROC GPLOT DATA=GRIDIF ;
PLOT F45M90*RAD S60M30*RAD E90M45*RAD / OVERLAY ;
```

B.3.5 TEST.DAT

```
       1.0000000000       1.0000000000
       1.0745699045       0.9306048828

 0.00000000  0.00000000 -1.00000000  0.00000000  0.00000000  0.00000000
 0.00000000 -1.00000000  0.00000000  0.00000000  0.00000000  1.00000000
 0.00000000  0.00000000  1.00000000  0.00000000  0.00000000  0.00000000

       0.000000000         1.000000000         0.000000000
       1.000000000        -4.000000000         1.000000000
       0.000000000         1.000000000         0.000000000

       0.000000000         1.000000000         0.000000000
       1.000000000         1.000000000         1.000000000
       0.000000000         1.000000000         0.000000000

-0.53728495 -0.93060488 -1.07456990  0.00000000 -0.53728495  0.93060488
 0.00000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000
 0.53728495 -0.93060488  1.07456990  0.00000000  0.53728495  0.93060488

       1.000000000         1.000000000         1.000000000
       0.000000000        -6.000000000         0.000000000
       1.000000000         1.000000000         1.000000000

       0.866025382         0.866025448         0.866025382
       0.000000000         0.866025404         0.000000000
       0.866025382         0.866025448         0.866025382

-1.00000000 -1.00000000 -1.00000000  0.00000000 -1.00000000  1.00000000
 0.00000000 -1.00000000  0.00000000  0.00000000  0.00000000  1.00000000
 1.00000000 -1.00000000  1.00000000  0.00000000  1.00000000  1.00000000

       1.000000000         1.000000000         1.000000000
       1.000000000        -8.000000000         1.000000000
```

1.000000000	1.000000000	1.000000000
0.500000000	1.000000000	0.500000000
1.000000000	0.750000000	1.000000000
0.500000000	1.000000000	0.500000000
12.211324880	11.674039928	11.211324880
21.150635120	20.220030237	-1.026447805
-0.523598754	-1.026447805	-1.094953004
0.000000000	-0.888929852	-0.547476502
12.211324880	12.211324880	12.211324880
20.150635120	21.150635120	-0.523598754
-0.963180127	-0.523598754	-0.963180127
-0.963180127	0.000000000	-0.963180127
12.211324880	12.748609832	13.211324880
21.150635120	20.220030237	-0.831400842
-0.523598754	-0.831400842	-0.735363290
0.000000000	-0.720014232	-0.367681645
11.211324880	11.136754976	11.211324880
21.150635120	21.150635120	-0.831400824
-0.815306521	-0.831400824	-0.815306521
-0.815306521	-0.720014271	-0.815306521
12.211324880	12.211324880	12.211324880
21.150635120	21.150635120	-0.523598754
-0.523598754	-0.523598754	-0.523598754
2.094395014	2.720698932	3.141592521
13.211324880	13.285894785	13.211324880
21.150635120	21.150635120	-0.022719093
-0.062227779	-0.022719093	-0.062227779
-0.062227779	-0.019675313	-0.062227779
12.211324880	11.674039928	11.211324880
21.150635120	22.081240003	-0.022719051
-0.523598754	-0.022719051	-0.171998288
0.000000000	-0.019675275	-0.085999144
12.211324880	12.211324880	12.211324880
22.150635120	21.150635120	-0.523598754
0.285591150	-0.523598754	0.285591150
0.285591150	0.000000000	0.285591150
12.211324880	12.748609832	13.211324880
21.150635120	22.081240003	0.461420901
-0.523598754	0.461420901	0.669396643
0.000000000	0.399602212	0.334698322

0.1465351279	0.1331823483	0.3175100285
0.0573681568	0.1196890743	0.2931383918
0.0891669711	0.0134932740	0.0243716367
12.500000000	11.962715048	11.500000000
21.650635095	20.720030212	-0.821327468
0.000000000	-0.821327468	-0.937891783
0.000000000	-0.711290434	-0.468945892
12.500000000	12.500000000	12.500000000
20.650635095	21.650635095	0.000000000
-0.710757432	0.000000000	-0.710757432
-0.710757432	0.000000000	-0.710757432
12.500000000	13.037284952	13.500000000
21.650635095	20.720030212	-0.479196873
0.000000000	-0.479196873	-0.317104900
0.000000000	-0.414996654	-0.158552450
11.500000000	11.425430096	11.500000000
21.650635095	21.650635095	-0.479196841
-0.451331773	-0.479196841	-0.451331773
-0.451331773	-0.414996659	-0.451331773
12.500000000	12.500000000	12.500000000
21.650635095	21.650635095	0.000000000
0.000000000	0.000000000	0.000000000
0.000000000	0.000000000	0.000000000
13.500000000	13.574569904	13.500000000
21.650635095	21.650635095	0.506089159
0.475339632	0.506089159	0.475339632
0.475339632	0.438286091	0.475339632
12.500000000	11.962715048	11.500000000
21.650635095	22.581239977	0.506089191
0.000000000	0.506089191	0.382901280
0.000000000	0.438286085	0.191450640
12.500000000	12.500000000	12.500000000
22.650635095	21.650635095	0.000000000
0.716479009	0.000000000	0.716479009
0.716479009	0.000000000	0.716479009
12.500000000	13.037284952	13.500000000
21.650635095	22.581239977	0.821327468
0.000000000	0.821327468	0.939767377
0.000000000	0.711290434	0.469883688
0.0799786871	0.0907083932	0.1066367643
0.0666899751	0.0865788623	0.1000200378

0.0132887121	0.0041295309	0.0066167265
12.788675120	12.251390168	11.788675120
22.150635069	21.220030186	-0.461420901
0.523598754	-0.461420901	-0.666089337
0.000000000	-0.399602212	-0.333044668
12.788675120	12.788675120	12.788675120
21.150635069	22.150635069	0.523598754
-0.276359838	0.523598754	-0.276359838
-0.276359838	0.000000000	-0.276359838
12.788675120	13.325960072	13.788675120
22.150635069	21.220030186	0.057289316
0.523598754	0.057289316	0.243383107
0.000000000	0.049614002	0.121691554
11.788675120	11.714105215	11.788675120
22.150635069	22.150635069	0.057289356
0.092038743	0.057289356	0.092038743
0.092038743	0.049614040	0.092038743
12.788675120	12.788675120	12.788675120
22.150635069	22.150635069	0.523598754
0.523598754	0.523598754	0.523598754
-2.094395014	-2.720698932	-3.141592521
13.788675120	13.863245024	13.788675120
22.150635069	22.150635069	0.846887027
0.829222172	0.846887027	0.829222172
0.829222172	0.733425717	0.829222172
12.788675120	12.251390168	11.788675120
22.150635069	23.081239952	0.846887045
0.523598754	0.846887045	0.774947539
0.000000000	0.733425676	0.387473770
12.788675120	12.788675120	12.788675120
23.150635069	22.150635069	0.523598754
0.966434783	0.523598754	0.966434783
0.966434783	0.000000000	0.966434783
12.788675120	13.325960072	13.788675120
22.150635069	23.081239952	1.026447805
0.523598754	1.026447805	1.096081854
0.000000000	0.888929852	0.548040927
0.2665990864	0.0242907566	0.1519827014
0.1718704035	0.0135500150	0.1186043801
0.0947286829	0.0107407416	0.0333783212

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 ΔU direction.

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

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

$$= \frac{f(U+2\Delta U) - 2f(U+\Delta U) + 2f(U-\Delta U) - f(U-2\Delta U)}{2(\Delta U)^3} \quad (C.1)$$

So that 3rd order difference convolution matrices can be produced by using the following vector within each

preferential direction of the digital filter.

$$\underset{1,5}{H'_3} = [\ -1 \quad +2 \quad 0 \quad -2 \quad +1\ ] \tag{C.2}$$

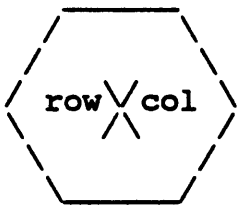
All unbiased convolutions using odd-ordered (1,3,5,...) differences will have zero in the digital filter's central element. Therefore, the construction of the convolution matrix that is using odd-ordered differences is easier because the preferential directions are not sharing the central pixel in the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

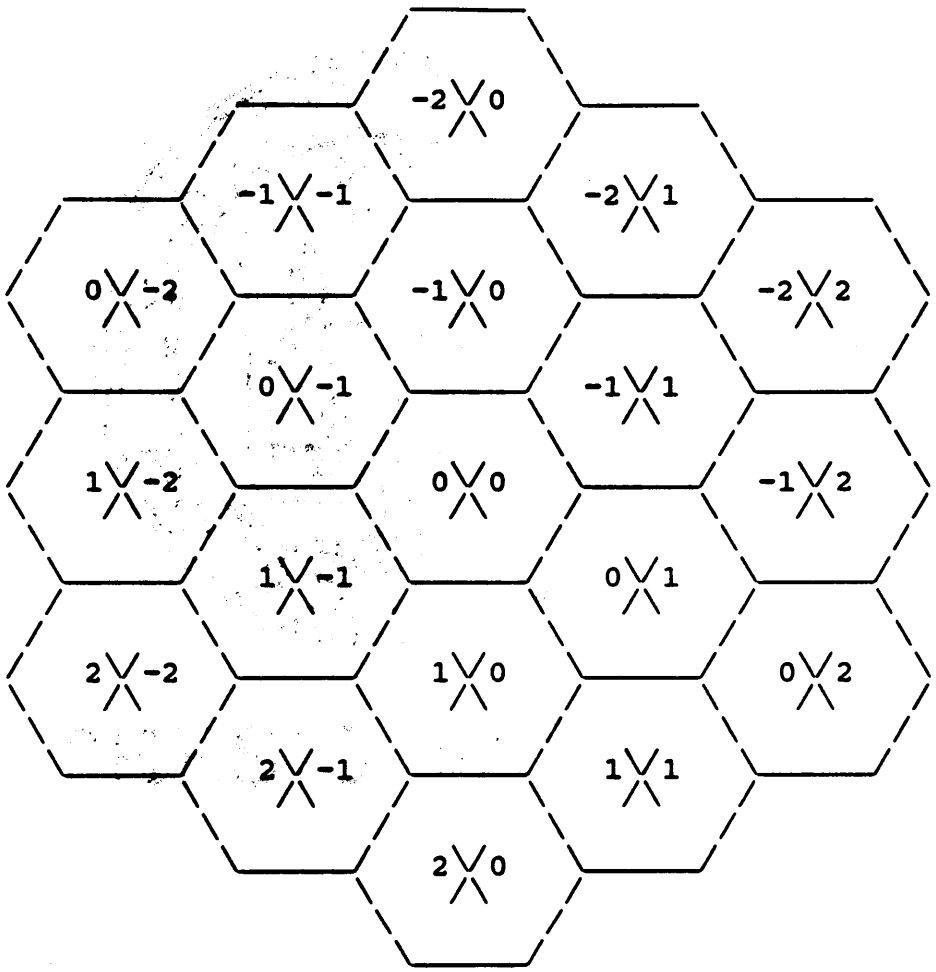


Figure C.1: 6-Neighbor Grid Lattice N=2 Window

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.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \Delta X & 0 \\ 0 & \Delta Y \end{bmatrix} \begin{bmatrix} r \\ c \end{bmatrix} \qquad \text{(C.3a)}$$

$$\begin{bmatrix} r \\ c \end{bmatrix} = \begin{bmatrix} \Delta X & 0 \\ 0 & \Delta Y \end{bmatrix}^{(-1)} \begin{bmatrix} x \\ y \end{bmatrix} \qquad \text{(C.3b)}$$

Where ΔX and ΔY are grid post spacings in the Figure 2.1 orthogonal $\underline{x}$ and $\underline{y}$ 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 $\underline{v}$ and $\underline{w}$ 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.

$$\begin{bmatrix} x \\ y \end{bmatrix} = [\Delta D] \begin{bmatrix} 1 & \sin 30^{o} \\ 0 & \cos 30^{o} \end{bmatrix} \begin{bmatrix} r \\ c \end{bmatrix} \qquad \text{(C.4a)}$$

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

Where ΔD 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).

$$\begin{bmatrix} (-1,-1) & (-1, 0) & (-1, 1) \\ (0,-1) & (0, 0) & (0, 1) \\ (1,-1) & (1, 0) & (1, 1) \end{bmatrix} \qquad \text{(C.5)}$$

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).

$$[\Delta D] \begin{bmatrix} \text{empty} & (-1, 0) & (-\sin 30^{o}, \cos 30^{o}) \\ (-\sin 30^{o}, -\cos 30^{o}) & (0, 0) & (\sin 30^{o}, \cos 30^{o}) \\ (\sin 30^{o}, -\cos 30^{o}) & (1, 0) & \text{empty} \end{bmatrix} \qquad \text{(C.6)}$$

where $\sin 30^{o} = 1 - \sin 30^{o}$. For this paper's purpose, the 6-neighbor grid lattice's elementary window storage array diagonal directions corresponded to the $\underline{u}$ and $\underline{w}$ preferential directions.

$$[\Delta D] \begin{bmatrix} (-\sin 30^{o}, -\cos 30^{o}) & (-1, 0) & (-\sin 30^{o}, \cos 30^{o}) \\ \text{empty} & (0, 0) & \text{empty} \\ (\sin 30^{o}, -\cos 30^{o}) & (1, 0) & (\sin 30^{o}, \cos 30^{o}) \end{bmatrix} \tag{C.7}$$

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 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.

$$\begin{bmatrix} - & - & + & + & + \\ - & + & + & + & + \\ + & + & + & + & + \\ + & + & + & + & - \\ + & + & + & - & - \end{bmatrix} \qquad \text{(C.8)}$$

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^{o} for the 6-neighbor grid lattice. These are numbers from the TEST.DAT file produced by write statements from the FILTER.FOR routine.

$$F'^{*}\ H_6 = \begin{bmatrix} -1.026447805 & -0.831400824 & -0.022719051 \\ & -0.523598754 & \\ -0.831400842 & -0.019675313 & 0.461420901 \end{bmatrix}$$

$$* \qquad \begin{bmatrix} 0.8666025 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ & -6 & \\ 1 & 1 & 1 \end{bmatrix}$$

$$= -0.888929852 - 0.720014271 - 0.019675275 + 2.720698932 - 0.720014232 - 0.019675313 + 0.399602212$$

$$= +0.751992201$$

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.

$$F'^{*}\ H_8 = \begin{bmatrix} -1.094953004 & -0.815306521 & -0.171998288 \\ -0.963180127 & -0.523598754 & 0.285591150 \\ -0.735363290 & -0.062227779 & 0.669396643 \end{bmatrix}$$

$$* \qquad \begin{bmatrix} 0.5 & 1.0 & 0.5 \\ 1.0 & -6.0 & 1.0 \\ 0.5 & 1.0 & 0.5 \end{bmatrix}$$

$$= -0.547476502 - 0.815306521 - 0.085999144 - 0.963180127 + 3.141592521 + 0.285591150 - 0.367681645 - 0.062227779 + 0.334698322$$

$$= +0.920010275$$

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

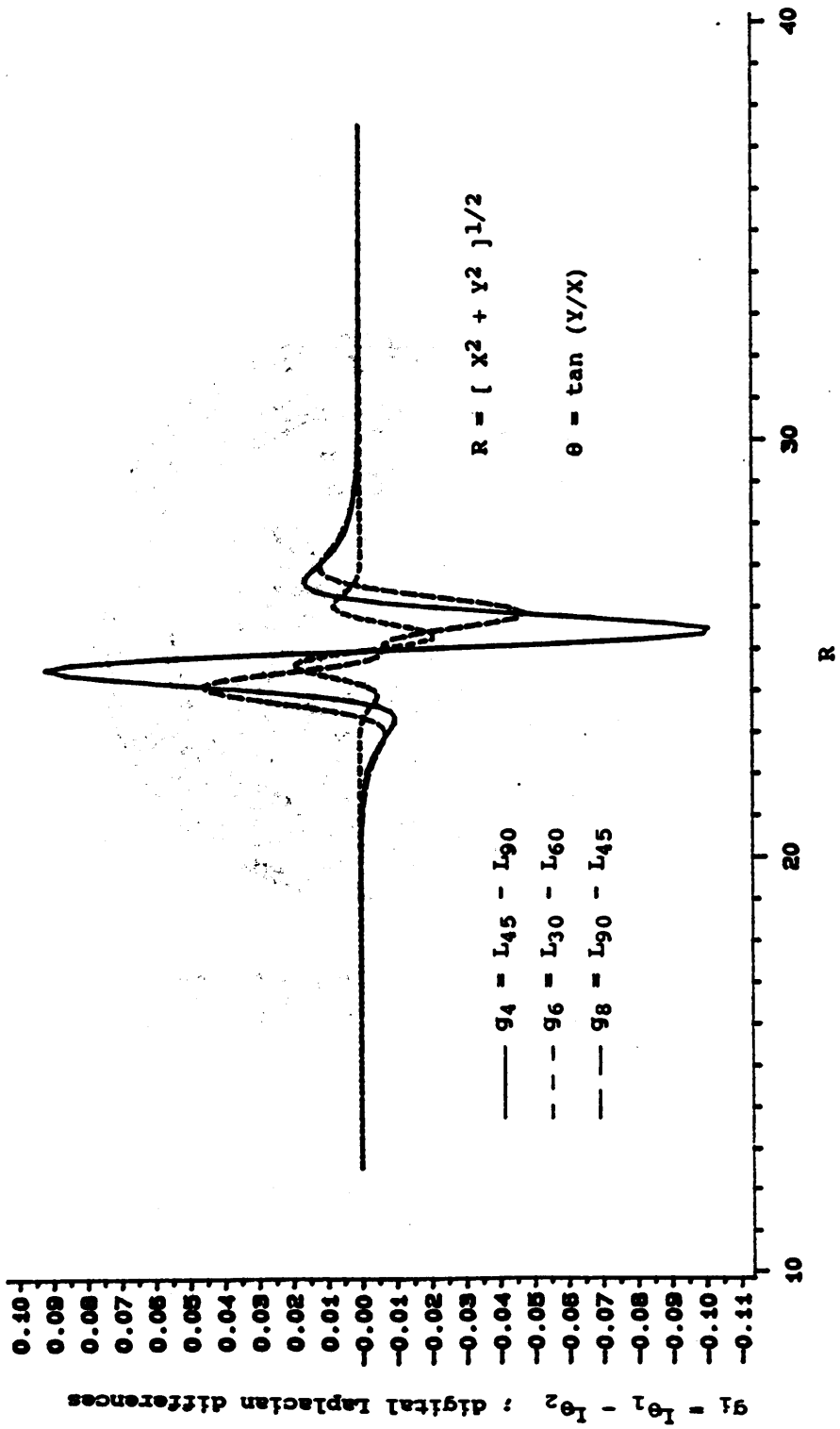


Figure C.2: Radial Profile Digital Laplacian Differences

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, hense 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.

$$g(x,y) = \int\int h(s)\, f(x-s,y-t)\, h(t)\, \delta s\, \delta t \qquad \text{(C.9a)}$$

$$g(X_i,Y_j) = \sum_t \sum_s h_x(i,s)\, f(X_{i+s},Y_{j+t})\, h_y(t,j) \qquad \text{(C.9b)}$$

if $h(s,t)=h(s)h(t)$. That interpretation produces vector products between the digital image and pairs of digital filters. Where $\underline{h}_x=[h(i,0),\ldots,h(i,m-1)]$ is the i^{th} row vector of the $\underline{H}_x$ matrix, that will produce vector products with all the $\underline{F}$ matrix's column vectors in the image's $\underline{x}$ direction. And where $\underline{h}_y=[h(0,j),\ldots,h(n-1,j)]^t$ is the j^{th}

column vector of the $\underline{H}_y$ matrix, that will produce vector products with all the $\underline{F}$ matrix's row vectors in the image's y direction. The $\underline{h}_x(i,s)$ and $\underline{h}_y(t,j)$ are vector elements that are discrete values of the h(s) and h(t) functions, respectively. So that if $\underline{F}$ is a matrix of (m,n) dimensions, then

$$\underset{m,n}{\underline{G}} = \underset{m,m}{\underline{H}_x} \underset{m,n}{\underline{F}} \underset{n,n}{\underline{H}_y} \qquad \text{(C.10)}$$

A vector $\underline{h}$ is periodic if:

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

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

The digital image $\underline{F}$ can be filtered by periodic vectors, if those vectors are contained as rows or columns of the $\underline{H}$ 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 $\underline{G}$ matrix to a $\underline{G}'$ matrix, accomplished by zeroing out portions of the $\underline{G}$ matrix recognized as paired vector products in two directions of $\underline{F}$, between vector pairs of various periodicities.

$$\underset{m,n}{\underline{F}'} = (\underset{m,m}{\underline{H}_x})^{-1} \underset{m,n}{\underline{G}'} (\underset{n,n}{\underline{H}_y})^{-1} \qquad (C.11)$$

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

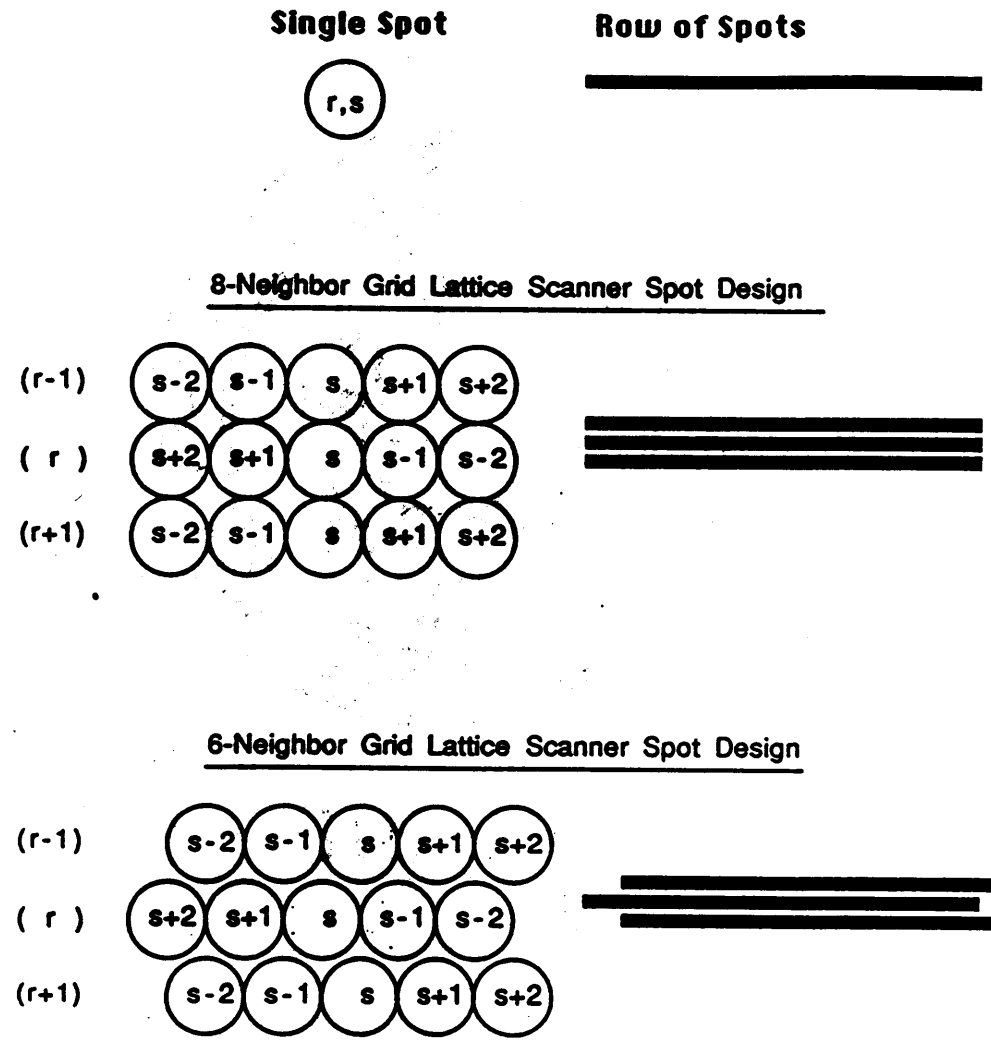
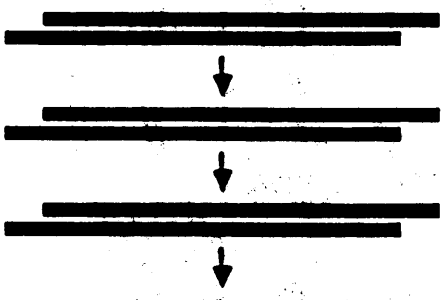


Figure C.3: Scanner Spot Designs

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)

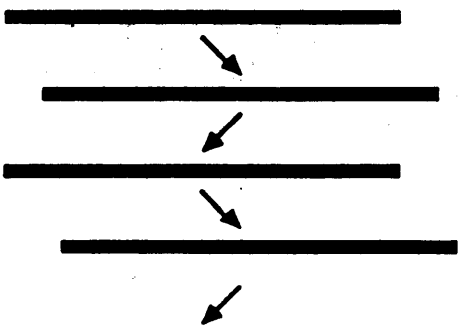


Figure C.4: 6-Neighbor Grid Lattice Scanning Patterns

The vita has been removed from the scanned document