

AFIT/GA/ENY/91D-14



# THE APPLICATION OF KRIGING FOR CONTROLLED MINIMIZATION OF LARGE DATA SETS



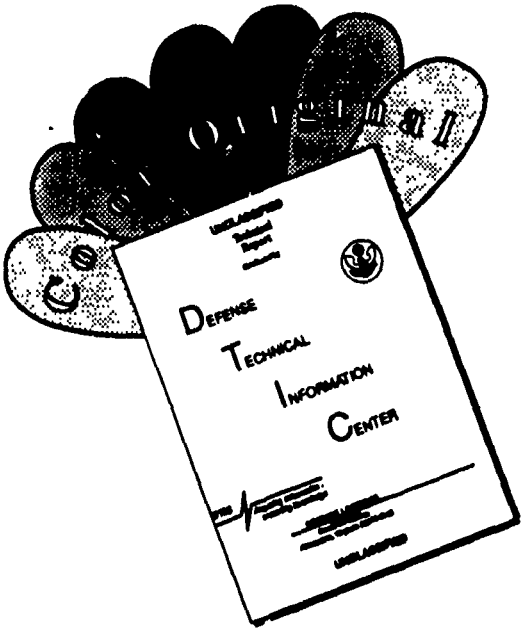
THESIS

Christopher Brodkin
Captain, USAF

AFIT/GA/ENY/91D-14



# DISCLAIMER NOTICE




THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

AFIT/GA/ENY/91D-14

# THE APPLICATION OF KRIGING FOR CONTROLLED MINIMIZATION OF LARGE DATA SETS

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

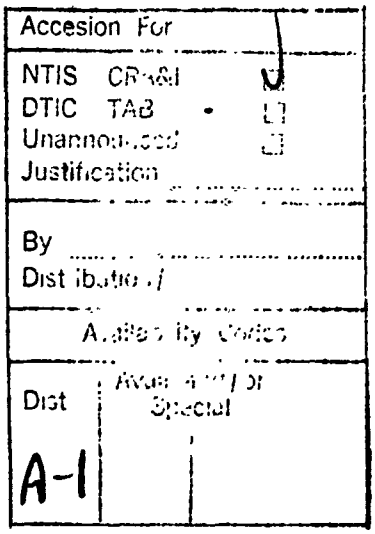
Requirements for the Degree of

Master of Science in Astronautical Engineering


Christopher Brodkin, B.S.

Captain, USAF

December 1991




Approved for public release; distribution unlimited

# *Preface*

This thesis develops the application of kriging for the controlled minimization of large data sets. As a result of this research, subsets of the original huge data set can be selected such that the largest estimation variance of the reconstructed image is within an acceptable level of error as defined by the user. The benefits of this effort will include drastically reduced on-line storage requirements, faster frame generation for computer animation, and topographical detail at greater or lesser resolution than that contained in the original data.

Special recognition is given to my advisor, Major David G. Robinson, who conceived and developed the idea of adapting the estimation technique of kriging from the field of geostatistics and applying it to the minimization of large data sets. I offer my personal thanks for his guidance and assistance in completing this research effort.

I would also like to thank the other members of my committee, Major Martin W. Stytz and Lt Col E. Phillip Amburn. Lt Col Amburn provided invaluable assistance on programming and optimal object structuring in C++.

This study was sponsored by the Human Engineering Division of the Harry G. Armstrong Laboratory.

Christopher Brodkin

# *Table of Contents*


| | | Page |
|---|---|---|
| Preface | | ii |
| Table of Contents | | iii |
| List of Figures | | vi |
| List of Tables | | ix |
| Abstract | | x |
| I. | Introduction | 1-1 |
| | 1.1 Background | 1-1 |
| | 1.2 Research Objective | 1-3 |
| | 1.3 Scope | 1-3 |
| | 1.3.1 Theory of Kriging | 1-3 |
| | 1.3.2 Procedural Development | 1-3 |
| | 1.3.3 Kriging Programs | 1-3 |
| II. | Literature Review | 2-1 |
| | 2.1 Structural Analysis | 2-1 |
| | 2.1.1 Trend Removal | 2-2 |
| | 2.1.2 Cluster Analysis | 2-2 |
| | 2.1.3 The Variogram | 2-2 |
| | 2.1.4 Standard Models | 2-4 |
| | 2.1.5 Problems with Anisotropy | 2-6 |
| | 2.2 Kriging | 2-7 |

Page

2.2.1 Origin of Kriging . . . . . 2-7

2.2.2 Definition . . . . . 2-8

2.2.3 Kriging Equations . . . . . 2-8

2.2.4 Kriging Assumptions . . . . . 2-11

2.2.5 Types of Kriging . . . . . 2-11

2.3 Data Set Minimization . . . . . 2-15

2.4 Summary . . . . . 2-16

III. Methodology . . . . . 3-1

3.1 Structural Analysis . . . . . 3-1

3.1.1 Trend Analysis . . . . . 3-1

3.1.2 Cluster Analysis . . . . . 3-2

3.1.3 The Semivariogram . . . . . 3-2

3.1.4 Data Set Reconstruction . . . . . 3-4

3.2 Kriging . . . . . 3-5

3.3 Minimization . . . . . 3-5

3.3.1 Background . . . . . 3-5

3.3.2 Szidarovszky's Methods . . . . . 3-7

3.3.3 Brodkin's Method . . . . . 3-9

3.4 Procedure Application . . . . . 3-16

IV. Results and Conclusions . . . . . 4-1

4.1 Results . . . . . 4-1

4.2 Conclusions . . . . . 4-14

V. Recommendations . . . . . 5-1

5.1 Structural Analysis . . . . . 5-1

5.2 Kriging . . . . . 5-1

5.3 Minimization . . . . . 5-2

Page

Bibliography . . . . . BIB-1

Vita . . . . . VITA-1

Appendix A. Users Manual . . . . . A-1

A.1 The Inputs . . . . . A-2

A.2 Example Krige Input Files . . . . . A-11

A.3 Example Varfit Input Files . . . . . A-13

Appendix B. Semivariogram Plots . . . . . B-1

Appendix C. Photographs . . . . . C-1

Appendix D. Minimized Image Tables . . . . . D-1

Appendix E. Programmers Manual . . . . . E-1

E.1 Matrix Object . . . . . E-2

E.2 Interface Routines . . . . . E-6

E.3 Kriging Routines . . . . . E-9

## *List of Figures*


| Figure | | Page |
|---|---|---|
| 2.1. | Experimental Semivariogram | 2-3 |
| 2.2. | Spherical Semivariogram | 2-5 |
| 2.3. | Linear Semivariogram | 2-5 |
| 2.4. | De Wijsian Semivariogram | 2-6 |
| 3.1. | Basic Kriging Pattern | 3-11 |
| 3.2. | Replicated Basic Kriging Pattern | 3-12 |
| 3.3. | Augmented Kriging Pattern | 3-13 |
| 3.4. | Kriging Areas | 3-13 |
| 3.5. | Partitioning of Slice 24 | 3-17 |
| 3.6. | Slice 12 Semivariogram in the $0^0$ Direction | 3-18 |
| 3.7. | Slice 12 Semivariogram in the $90^0$ Direction | 3-19 |
| 4.1. | Slice 1 Semivariogram in the $0^0$ Direction | 4-3 |
| 4.2. | Slice 1 Semivariogram in the $90^0$ Direction | 4-4 |
| 4.3. | Slice 9, Maximum Variance = 70 | 4-6 |
| 4.4. | Slice 24, Maximum Variance = 70 | 4-7 |
| 4.5. | Slice 9, Semi-Inclusion Distances = (1,1) | 4-8 |
| 4.6. | Slice 24, Semi-Inclusion Distances = (1,1) | 4-9 |
| 4.7. | Slice 24 Partitioning | 4-10 |
| 4.8. | Slice 24 Partitioned Image | 4-11 |
| B.1. | Slice 1 Semivariogram in the $0^0$ Direction | B-2 |
| B.2. | Slice 1 Semivariogram in the $90^0$ Direction | B-3 |
| B.3. | Slice 3 Semivariogram in the $0^0$ Direction | B-4 |

Figure Page

B.4. Slice 3 Semivariogram in the $90^0$ Direction . . . . . B-5

B.5. Slice 6 Semivariogram in the $0^0$ Direction . . . . . B-6

B.6. Slice 6 Semivariogram in the $90^0$ Direction . . . . . B-7

B.7. Slice 9 Semivariogram in the $0^0$ Direction . . . . . B-8

B.8. Slice 9 Semivariogram in the $90^0$ Direction . . . . . B-9

B.9. Slice 12 Semivariogram in the $0^0$ Direction . . . . . B-10

B.10. Slice 12 Semivariogram in the $90^0$ Direction . . . . . B-11

B.11. Slice 15 Semivariogram in the $0^0$ Direction . . . . . B-12

B.12. Slice 15 Semivariogram in the $90^0$ Direction . . . . . B-13

B.13. Slice 18 Semivariogram in the $0^0$ Direction . . . . . B-14

B.14. Slice 18 Semivariogram in the $90^0$ Direction . . . . . B-15

B.15. Slice 21 Semivariogram in the $0^0$ Direction . . . . . B-16

B.16. Slice 21 Semivariogram in the $90^0$ Direction . . . . . B-17

B.17. Slice 24 Semivariogram in the $0^0$ Direction . . . . . B-18

B.18. Slice 24 Semivariogram in the $90^0$ Direction . . . . . B-19

B.19. Slice 24 Center Partition Semivariogram in the $0^0$ Direction . . . B-20

B.20. Slice 24 Center Partition Semivariogram in the $90^0$ Direction . . . B-21

B.21. Slice 24 Background Partition Semivariogram in the $0^0$ Direction B-22

B.22. Slice 24 Background Partition Semivariogram in the $90^0$ Direction B-23

C.1. Slice 1, Maximum Variance = 100, 85, 70 . . . . . C-2

C.2. Slice 1, Semi-Inclusion Distance = 5, 4, 3 . . . . . C-3

C.3. Slice 1, Semi-Inclusion Distance = 3, 2, 1 . . . . . C-4

C.4. Slice 3, Maximum Variance = 100, 85, 70 . . . . . C-5

C.5. Slice 3, Semi-Inclusion Distance = 5, 4, 3 . . . . . C-6

C.6. Slice 3, Semi-Inclusion Distance = 3, 2, 1 . . . . . C-7

C.7. Slice 6, Maximum Variance = 100, 85, 70 . . . . . C-8

C.8. Slice 6, Semi-Inclusion Distance = 5, 4, 3 . . . . . C-9

Figure Page

C.9. Slice 6, Semi-Inclusion Distance = 3, 2, 1 . . . . . C-10

C.10.Slice 9, Maximum Variance = 100, 85, 70 . . . . . C-11

C.11.Slice 9, Semi-Inclusion Distance = 5, 4, 3 . . . . . C-12

C.12.Slice 9, Semi-Inclusion Distance = 3, 2, 1 . . . . . C-13

C.13.Slice 12, Maximum Variance = 100, 85, 70 . . . . . C-14

C.14.Slice 12, Semi-Inclusion Distance = 5, 4, 3 . . . . . C-15

C.15.Slice 12, Semi-Inclusion Distance = 3, 2, 1 . . . . . C-16

C.16.Slice 15, Maximum Variance = 100, 85, 70 . . . . . C-17

C.17.Slice 15, Semi-Inclusion Distance = 5, 4, 3 . . . . . C-18

C.18.Slice 15, Semi-Inclusion Distance = 3, 2, 1 . . . . . C-19

C.19.Slice 18, Maximum Variance = 100, 85, 70 . . . . . C-20

C.20.Slice 18, Semi-Inclusion Distance = 5, 4, 3 . . . . . C-21

C.21.Slice 18, Semi-Inclusion Distance = 3, 2, 1 . . . . . C-22

C.22.Slice 21, Maximum Variance = 100, 85, 70 . . . . . C-23

C.23.Slice 21, Semi-Inclusion Distance = 5, 4, 3 . . . . . C-24

C.24.Slice 21, Semi-Inclusion Distance = 3, 2, 1 . . . . . C-25

C.25.Slice 24, Maximum Variance = 100, 85, 70 . . . . . C-26

C.26.Slice 24, Semi-Inclusion Distance = 5, 4, 3 . . . . . C-27

C.27.Slice 24, Semi-Inclusion Distance = 3, 2, 1 . . . . . C-28

C.28.Slice 24, Partitioned, Maximum Variance = 13, 12, 11 . . . . . C-29

E.1. Matrix Object Structure . . . . . E-5

# *List of Tables*


| Table | | Page |
|---|---|---|
| 2.1. | Water Table Elevation Data | 2-12 |
| 3.1. | Slice 1 Statistics | 3-21 |
| 4.1. | Slice 9 Statistics | 4-13 |
| 4.2. | Slice 24 Statistics | 4-13 |
| D.1. | Slice 1 Statistics | D-2 |
| D.2. | Slice 3 Statistics | D-2 |
| D.3. | Slice 6 Statistics | D-2 |
| D.4. | Slice 9 Statistics | D-3 |
| D.5. | Slice 12 Statistics | D-3 |
| D.6. | Slice 15 Statistics | D-3 |
| D.7. | Slice 18 Statistics | D-4 |
| D.8. | Slice 21 Statistics | D-4 |
| D.9. | Slice 24 Statistics | D-4 |
| D.10. | Slice 24, Partitioned, Statistics | D-5 |

## *Abstract*

Frequently, the quantity of data available is much greater than that which can be manipulated in an efficient and timely manner. This can cause several problems. The first, and probably most critical, problem is the excessive on-line storage needs of these huge data sets. Secondly, in the computer animation field, huge data sets may require excessive computational time for generation of each frame of a computer animation. Thirdly, computer screens have a limited resolution and need too much computational time removing excessive detail from images generated with a higher resolution than can be displayed. Lastly, too much time is required to transmit huge amounts of data from location to location. What is needed is a method of minimizing the data set based on some acceptable level of resolution.

This thesis develops the application of kriging for the controlled minimization of large data sets based on a maximum acceptable level of error. Specifically, the geostatistical estimation technique of kriging is used to produce minimal data sets and to estimate the unknown values on an arbitrarily sized grid using as input any data set. All the procedures necessary to improve the accuracy of the estimate as well as the kriging procedure are developed. Using these procedures the entire process can be automated. The techniques are demonstrated using Magnetic Resonance Image data to support minimizing on-line storage requirements. Two concurrent thesis efforts also use the techniques to enlarge satellite photographs and to change the grid resolution of terrain data.

# THE APPLICATION OF KRIGING FOR CONTROLLED MINIMIZATION OF LARGE DATA SETS

## *I. Introduction*

This research effort develops the application of kriging to the controlled minimization of large data sets. Kriging is a geostatistical estimation procedure named after D.G. Krige, a South African mining engineer. Although kriging has its origins in geostatistics, the methods are applicable to a wide range of disciplines. This first chapter provides the background, the research objectives, and the scope of the study.

### *1.1 Background*

Historically, man has had more data than he can manipulate in an efficient and timely manner. The quantity of data that can be manipulated has grown dramatically, but the quantity available has grown even faster. What is needed is a method of choosing a minimal data set based on the desired level of acceptable error in the data needed for the application. Using a subset of the data will then allow faster manipulation of that data while maintaining the required level of acceptable error in the data.

The three areas that will probably benefit most from the use of kriging are computer graphics, computer animation, and topographical estimation. These were the main focus of this kriging development effort. All three of these areas require the ability choose a subset of the original larger data set and estimate the unknown points based on that minimal data set. For enlargement of the original data set the subset is equal to the original data set.

In computer graphics, an image may be displayed on the entire screen or it may displayed on only a very small portion of the screen. Using the entire data set

in both cases may waste substantial computer time calculating points that can't be displayed on the small portion of the screen. Using a subset of the entire data set in both cases may discard substantial detail needed for the entire screen display of the image. This problem can be alleviated by having multiple subsets of the original data set, one for each range of resolutions displayed. A range of resolutions would be used to minimize the number of subsets needed.

In computer animation data sets containing large numbers of graphics primitives (points, lines, polygons, patches) require great amounts of time for generation of each frame of the animation. In this case what is needed is a minimal data set based on the desired resolution of the animated object. The smaller the animated object is relative to the overall image size the fewer graphics primitives needed in the minimal data set. Faster moving objects also require fewer graphics primitives in the minimal data set. Therefore, in selecting a minimal data set, smaller or faster moving animated objects require fewer graphics primitives in the minimal data set than larger or slower moving objects.

Part of this approach involves estimating points that aren't known because they were removed while producing a minimal data set or were not known in the first place. This aspect of the method can then be used for topographical estimation. A photograph can be enlarged by estimating points between known points of the original image. This does not add detail but can aid a photo-interpreter in discerning more information from the enlarged image. This aspect can also be used to generate biofidelic head forms to assist in the design of environmental protection equipment. This involves estimating a general head shape, using many head shape data sets, thereby minimizing the variability between the different head shapes.

All these uses minimize the amount of data that must be readily available on immediate access storage devices. The large original data sets can be stored on slow access, long term storage devices.

## *1.2 Research Objective*

The purpose of this research effort was to develop a procedure that could select a minimal data set in a statistically controlled fashion and estimate unknown values, at regular grid locations, based on a desired overall maximum acceptable level of error.

## *1.3 Scope*

This thesis develops and demonstrates the application of kriging in the controlled minimization of large data sets. Specifically, this study includes a discussion of the theoretical development of kriging and the computer programs necessary for the application of the technique. The following provides a summary of the extent of this research effort.

*1.3.1 Theory of Kriging* The literature review provides an introduction to the theory and the development of kriging from the field of geostatistics. Emphasis is placed on the kriging equations and structural analysis of the data.

*1.3.2 Procedural Development* A complete development of the kriging procedure is outlined in Chapter III.

*1.3.3 Kriging Programs* This document includes a complete package of the programs required for the kriging of data. Specifically, the following programs, all written in C++, are included:

*Zonal Trend Partitioning Program.* This partitions the data based on variations of the data from the row and column median values of the data. This program isolates the zonal anisotropic behavior of the data into data subsets.

*Global Trend Removal Program.* This determines the least squares coefficients for a polynomial surface through the points and outputs the residuals.

*Least Squares Semivariogram Estimation Program.* This determines the least squares parameters for three of the more commonly used semivariogram models: the linear, the De Wijsian, and the spherical models.

*Kriging Program.* This program selects the minimal data set and/or estimates the values at the grid locations and provides the kriging variances for these points.

*Global Trend Addition Program.* This uses the coefficients for a polynomial surface through the points to add the global trend to the kriged residuals and outputs the estimates.

# *II. Literature Review*

The first step is to research the current methods pertaining to the application of kriging and the structural analysis needed for kriging. Therefore, this chapter provides a review of the literature pertaining to the areas of kriging and structural analysis. The emphasis of this review is on the development of kriging in the field of geostatistics and the structural analysis essential to the application of the kriging procedures.

## *2.1 Structural Analysis*

The goal of structural analysis, as it pertains to using the kriging method, is to determine the spatial distribution of the quantity of interest, the semivariogram. Before characterizing the spatial distribution, however, the presence of global trend must be determined and, if present, removed from the data. In kriging, the spatial distribution of the quantity of interest is characterized by the semivariogram. The semivariogram is a function describing the expected difference in value between pairs of samples with a given spatial relationship (4:11).

Yakowitz and Szidarovszky note:

> The kriging method is composed of two activities, (i) inferring the semivariogram from the data and (ii) assuming that the inferred semivariogram is indeed exact, providing a best linear unbiased estimator and associated error variance. (20:23–24)

Journel and Huijbregts emphasize that the first and most important step in any geostatistical study is structural analysis (12:12). "Structural analysis is the name given to the procedure of characterizing the structures of the spatial distribution of the variables considered (e.g., grades, thicknesses, accumulations)" (12:12).

*2.1.1 Trend Removal* The goal of computing and using the semivariogram is to improve the accuracy of the estimate of the desired quantity at the desired location. To achieve this goal, it is necessary to remove any trend in the data. In this context trend is the tendency for the local mean to increase, or decrease, as a function of spatial location in the data. One method to remove the trend is to fit a polynomial to the data and then subtract the value of the polynomial at each point from the known value at that point. This method provides a continuous trend value for any point and, for this research effort, is preferred over the discrete method of median polishing as suggested by Cressie (5). The problem with discrete methods is that the trend is only known at the original data sites. Therefore, the trend can only be estimated at the desired location and added to the estimate of the residual for that location.

*2.1.2 Cluster Analysis* Another method to improve the accuracy of the estimate is to perform cluster analysis on the data before calculating the semivariogram. Cluster analysis refers to a number of techniques which classify objects in homogeneous and distinct groups (1). The definition of a cluster is often determined by the researcher. The goal, however, is to partition the original data set into subsets that contain some degree of similarity. The method chosen is to calculate the row and column sums, find the medians of the row and column sums, and then partition the data based on row and column sums that fall below, or above, the corresponding row or column median.

*2.1.3 The Variogram* The variogram is twice the semivariogram and according to Omre, "the variogram function is the backbone of geostatistical analysis" (15:107). The semivariogram function is defined as the variance of the difference of the quantity of interest between two points, $[Z(x_1, y_1) - Z(x_2, y_2)]$. Using this definition, it can be seen that the semivariogram is zero between a point and itself.

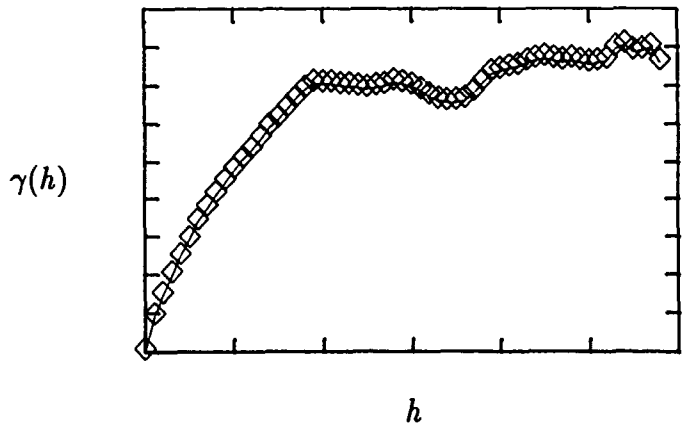



Figure 2.1. Experimental Semivariogram

The variogram function is written as:

$$2\gamma(h_{12}) = Var\{Z(x_1, y_1) - Z(x_2, y_2)\}$$

Where $h_{12}$ is the distance between the points $(x_1, y_1)$ and $(x_2, y_2)$. The semivariogram is simply $\gamma(h_{12})$.

In practice, only an *estimator* of the theoretical semivariogram is available. This estimator is known as the experimental semivariogram and is calculated as follows:

$$\gamma^*(h) = \frac{1}{2|N|} \sum_{i=1}^{|N|} [z(x_i + h) - z(x_i)]^2$$

where $|N|$ is the number of pairs of data values at a distance of $h$ apart from one another, $x_i$ is the location of point $i$, $x_i + h$ is the location of a point at distance $h$ from $i$, and $z(x_i)$ and $z(x_i + h)$ are the values of the quantity of interest at $i$ and $x_i + h$. An example of an experimental semivariogram is shown in Figure 2.1.

The next step in the structural analysis procedure is to fit a model to the experimental semivariogram so that semivariogram values may be calculated for the points to be estimated. The approach taken is to calculate the parameters for each

model using a numerical least-squares fitting routine and then select the model with the best simple correlation. Since the number of pairs of points $h$ distance apart used in calculating the semivariogram decreases as $h$ increases Cressie proposes minimizing a weighted sum of squares and indicates that work by Zimmerman and Zimmerman shows that the weighted least squares approach never performs poorly and usually does well (5:198). Clark suggests limiting $h$ to half the largest distance in the data set (4:14).

*2.1.4 Standard Models* Three of the more common models are the spherical model, the linear model, and the De Wijsian model (6:120- 122). A brief introduction to each of these models is provided.

*Spherical Model* (6:80) The spherical model is the most common model and is defined by three parameters: $a$, $C$, and $C_0$. The first parameter, $a$, is called the range and is used to determine the range of influence. The third parameter, $C_0$, is known as the nugget effect. Finally, the second parameter, $C$, is used in conjunction with $C_0$ to determine the sill, $(C+C_0)$, defined as the covariance of the samples at $h$=0.0, $\sigma(0)$. The form of the spherical model is as follows:

$$\gamma(h) = \begin{cases} C(\frac{3}{2}\frac{h}{a} - \frac{1}{2}\frac{h^3}{a^3}) + Co & \text{if } h < a \\ C + C_0 & \text{if } h \geq a \\ 0 & \text{if } h = 0 \end{cases}$$

The shape of this model is shown in Figure 2.2.

*Linear Model* The equation for the linear model is of the form $\gamma(h) = ah + b$. This is one of two models used in practice which does not have a sill (6:120). The shape of this model is shown in Figure 2.3.

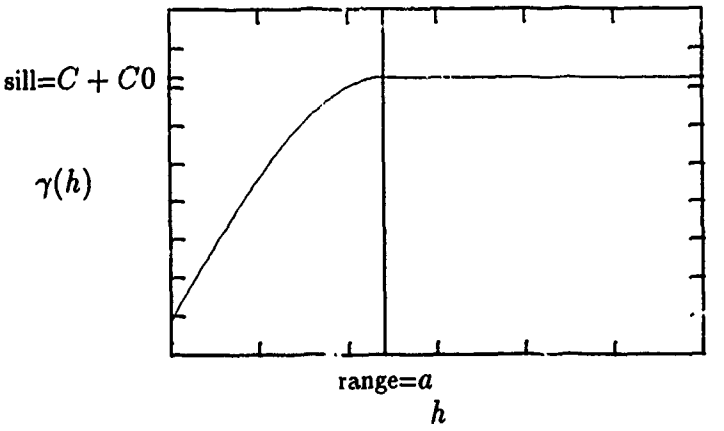



Figure 2.2. Spherical Semivariogram

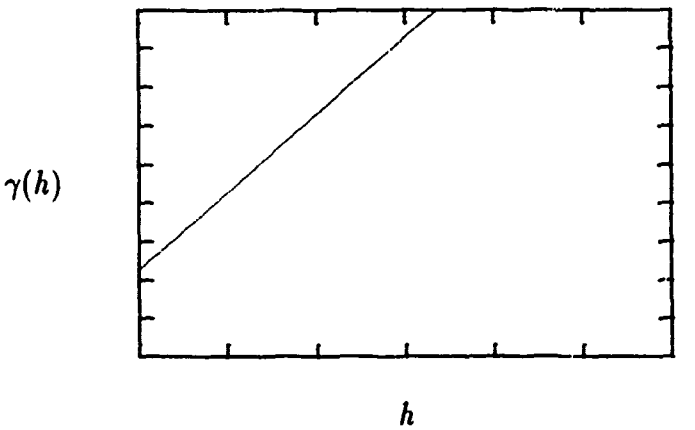



Figure 2.3. Linear Semivariogram

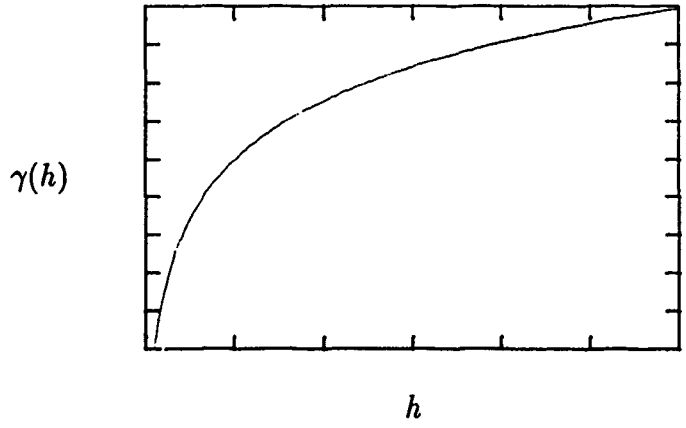



Figure 2.4. De Wijsian Semivariogram

*Dc Wijsian Model* The form for the De Wijsian model is $\gamma(h) = a\ln(h)+b$. However, "one usually writes $a = 3\alpha$ and calls it the coefficient of intrinsic dispersion" (6:121). This model also does not have a sill. This model is named after Prof. H.J. de Wijs and is used when the experimental data plots as a straight line on a logarithmic scale (6:120). The shape of this model is shown in Figure 2.4.

*2.1.5 Problems with Anisotropy* Anisotropies are typically classified in one of two categories: geometric and zonal (6:134). Geometric anisotropy refers to the situation where the value or expected variation varies more quickly in one direction than in another. This is evidenced by different semivariogram ranges in different directions but, for the spherical model, identical sills. This type of anisotropy can be handled by scaling the coordinates of the data sets or by using different semivariograms for different directions. The method chosen is to scale the coordinates of the data sets. Zonal isotropy is characterized by qualitative variations or separations of the data into zones. This form is very difficult to treat. In the spherical model this is evidenced by different sills in different directions.

The anisotropy ratio (or affinity modulus), $k$, is equal to the ratio of the ranges

in each direction, $k = a(y)/a(x)$. For example, if the range is 50 feet in the x direction and 300 feet in the y direction, then $k$ is equal to 300/50 (6:134–135). The distance vector, $\vec{h}$, can be decomposed into two components, $h_1 = (x_1 - x_2)$ along the x axis, and $h_2 = (y_1 - y_2)$ along the y axis. Therefore, using the semivariogram model calculated in the $y$ direction, the scaled distance between the two points $(x_1, y_1)$ and $(x_2.y_2)$ is $h = \sqrt{k^2(x_1 - x_2)^2 + (y_1 - y_2)^2}$. This treatment of geometric anisotropy can be extended to three dimensions by introducing a second anisotropy ratio for the third direction, $c = a(y)/a(z)$, and multiplying it by the change in distance in that direction. Treating zonal anisotropy is beyond the scope of this study; for more information reference David (6:134–148).

## *2.2 Kriging*

A review of the literature showed some applications of kriging in the controlled minimization of large data sets. In particular, Ferenc Szidarovszky explains how a minimal data set can be constructed by testing all possible subsets of the data and using a branch and bound technique to reduce the time required for this testing (18). This review explains the theory of the technique as developed in the field of geostatistics and applied to this effort. Specifically, the following kriging topics are discussed: the origin, a definition, the fundamental equations, the universal equations, the assumptions, and several types of kriging.

*2.2.1 Origin of Kriging* The method of kriging traces its origins to the field of geostatistics as developed in the mining industry. According to Journel: "in mining practice, one problem is to find the best possible estimator of the mean grade of a block" (11:563). He further states that D.G. Krige proposed a regression technique for this problem in 1951 and that "in 1963, Matheron formalized and generalized this regression procedure and gave it the name of kriging" (11:563) after D.G. Krige, a mining engineer in South Africa. Georges Matheron is also credited with introducing the concept of regionalized variables. According to Matheron:

> Geostatistics, in their most general acceptation, are concerned with the study of the distribution in space of useful values for mining engineers and geologists, such as grade, thickness, or accumulation, including a most important practical application of the problems arising in ore-deposit evaluation. (13:224)

*2.2.2 Definition* Matheron originally defined kriging as follows: "kriging is the probabilistic process of obtaining the best linear unbiased estimator of an unknown variable" (11:563). In this context, "best" is defined as "having the smallest estimation variance" (4:104). Matheron later generalized the techniques for obtaining nonlinear unbiased estimates. Journel states that kriging should be redefined as "a probabilistic theory of estimation based on the principle of minimization of the estimation variance" (11:563). Therefore, kriging is a method of estimating an unknown value at a point based on known values of surrounding points with the constraint that the estimation error is minimized.

*2.2.3 Kriging Equations* The estimate for an unknown value at a point is the weighted average of surrounding values with the closer points having more weight than points further away. Specifically, the equation for the estimator is:

$$X_0 = w_1X_1 + w_2X_2 + w_3X_3 + \ldots + w_nX_n$$

where $X_0$ is the estimate, $w_1$, $w_2$, $w_3$, ..., $w_n$ are the weights, and $X_1$, $X_2$, $X_3$, ..., $X_n$ are the sample values (4:99).

For the estimator to be unbiased the weights must sum to one and there must be no trend. An unbiased estimator is one in which, over a large number of estimations, the average error is zero. Trend is the tendency for the local mean to increase, or decrease, as a function of spatial location in the data. To determine the weights that minimize the estimation variance, the estimation variance must be defined. The

estimation variance of $X_0$ for the general unbiased linear estimator is:

$$Var(X_0 - X) = \sum_{i=1}^{n} w_i \gamma(h_{ip}) - \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \gamma(h_{ij})$$

where $X_0 - X$ is the estimation error, $Var(X_0 - X)$ is the variance of this error, $w_i$ and $w_j$ are the weights, $\gamma(h_{ip})$ is the semivariogram value between the value being estimated and the known value at point $i$, and $\gamma(h_{ij})$ is the semivariogram value between the known value at point $i$ and the known value at point $j$. The semivariogram was discussed in the structural analysis section of this review. For any given set of observations the variance is a function only of the values of the weights. Therefore, to minimize the estimation variance, the partial derivatives of the estimation variance with respect to the weights must be set to zero and the weights must be determined by solving the resulting system of equations. The result is the following system of linear equations (referred to as the kriging system):

$$\begin{array}{ccccccccccccc}
0 & + & w_1 & + & w_2 & + & w_3 & + & \cdots & + & w_n & = & 1 \\
\lambda & + & w_1\gamma(h_{11}) & + & w_2\gamma(h_{12}) & + & w_3\gamma(h_{13}) & + & \cdots & + & w_n\gamma(h_{1n}) & = & \gamma(h_{1p}) \\
\lambda & + & w_1\gamma(h_{21}) & + & w_2\gamma(h_{22}) & + & w_3\gamma(h_{23}) & + & \cdots & + & w_n\gamma(h_{2n}) & = & \gamma(h_{2p}) \\
\lambda & + & w_1\gamma(h_{31}) & + & w_2\gamma(h_{32}) & + & w_3\gamma(h_{33}) & + & \cdots & + & w_n\gamma(h_{3n}) & = & \gamma(h_{3p}) \\
\vdots & + & \vdots & + & \vdots & + & \vdots & + & \vdots & + & \vdots & = & \vdots \\
\lambda & + & w_1\gamma(h_{n1}) & + & w_2\gamma(h_{n2}) & + & w_3\gamma(h_{n3}) & + & \cdots & + & w_n\gamma(h_{nn}) & = & \gamma(h_{np})
\end{array}$$

As stated previously, to maintain the unbiased nature of the estimate, the weights must sum to one. A Lagrangian Multiplier ($\lambda$) is used so that the number of unknowns and the number of equations are equal. Using the calculated weights the minimized estimation variance of the estimate, $X_0$, is:

$$Var(X_0 - X) = \sum_{i=1}^{n} w_i \gamma(h_{ip}) + \lambda$$

It is important to notice that the weights and the Lagrangian Multiplier depend

only on the semivariogram and therefore the estimation variance depends only on the semivariogram. As shown in the structural analysis section, the semivariogram is a function of the relative distance and orientation of the known points, the pattern. Therefore, the estimation variance depends only on the pattern of the points, particularly the distances between the known points and the distances between the point to be estimated and the known points, and not the values at these known points. This fact will be very useful later in the procedure development.

Following a similar development using the covariance function, which is a function of the semivariogram as follows: $\sigma(h_{ij}) = \sigma(0) - \gamma(h_{ij})$, an equivalent form of the kriging equations can be derived as follows:

$$
\begin{array}{ccccccccccccc}
0 & + & w_1 & + & w_2 & + & w_3 & + & \ldots & + & w_n & = & 1 \\
\lambda & + & w_1\sigma(h_{11}) & + & w_2\sigma(h_{12}) & + & w_3\sigma(h_{13}) & + & \cdots & + & w_n\sigma(h_{1n}) & = & \sigma(h_{1p}) \\
\lambda & + & w_1\sigma(h_{21}) & + & w_2\sigma(h_{22}) & + & w_3\sigma(h_{23}) & + & \cdots & + & w_n\sigma(h_{2n}) & = & \sigma(h_{2p}) \\
\lambda & + & w_1\sigma(h_{31}) & + & w_2\sigma(h_{32}) & + & w_3\sigma(h_{33}) & + & \cdots & + & w_n\sigma(h_{3n}) & = & \sigma(h_{3p}) \\
\vdots & + & \vdots & + & \vdots & + & \vdots & + & \vdots & + & \vdots & = & \vdots \\
\lambda & + & w_1\sigma(h_{n1}) & + & w_2\sigma(h_{n2}) & + & w_3\sigma(h_{n3}) & + & \ldots & + & w_n\sigma(h_{nn}) & = & \sigma(h_{np})
\end{array}
$$

Both forms of the equations can be put into the matrix form $[A]\cdot\{X\} = \{B\}$, where [A] is the square matrix of the semivariogram (or covariance) values between the known points, {X} is the column vector of weights, and {B} is the column vector of the semivariogram (or covariance) values between the point being estimated and the known points. In matrix form, the solution is found by inverting [A] and post-multiplying by {X}. As shown in the structural analysis section, the value of the semivariogram between the point and itself is zero but the covariance is not. What this means is that [A], in the semivariogram form, has a zero for all the diagonal terms but the covariance form does not. This makes the covariance form more stable during matrix inversion than the semivariogram form. For this reason the covariance form of the equations is preferred. However, since the covariance does not exist for all

theoretical semivariogram models, both forms of the equations are developed in this thesis. The minimized estimation variance of the estimate, $X_0$, for the covariance form is:

$$Var(X_0 - X) = \sigma(0) - \sum_{i=1}^{n} w_i \sigma(h_{ip}) - \lambda$$

*2.2.4 Kriging Assumptions* The underlying assumption of kriging is that the values fall within some probability distribution, usually a normal or lognormal distribution. This assumption allows statistical methods to be applied to the data. Kriging usually assumes some form of stationarity. The most stringent form is strong stationarity in which all higher order moments exist and are constant everywhere over the data field. The second form is weak stationarity. Weak stationarity implies that all random variables have the same mean, variance and autocorrelation function. This assumption is based on two conditions: 1) the expected value of the regionalized variable is the same all over the field of interest; and, 2) the spatial covariance of the regionalized variable is the same all over the field of interest (6:92). The assumption of weak stationarity can be further relaxed by requiring only that all random variables in a subsection of the data field, the neighborhood, have the same first and second order moments after removal of any local drift. This is the assumption used for universal kriging.

*2.2.5 Types of Kriging* The more popular types of kriging are point, block, lognormal, disjunctive, universal kriging, and cokriging. "The kriging techniques are all related, and are refined versions of the weighted moving average techniques used by Krige" (10:25). Point and universal kriging are central to this effort and are discussed below in more detail.

*Point Kriging.* The system of kriging equations previously developed were those specific to point kriging. Davis discusses this simplest form of kriging and provides an example to illustrate the mechanics of the kriging system.

Table 2.1. Water Table Elevation Data

| Location | X Coordinate | Y Coordinate | Water Table Elevation |
|---|---|---|---|
| Well 1 | 3.0 | 4.0 | 120.0 |
| Well 2 | 6.3 | 3.4 | 103.0 |
| Well 3 | 2.0 | 1.3 | 142.0 |
| Point p | 3.0 | 3.0 | |

The following example uses the semivariogram form of the kriging equations and is adapted from *Statistics and Data Analysis in Geology* and demonstrates the use of kriging in estimating the water elevation at an unsampled location (7:386–390).

The basic problem is to estimate the water elevation at some point $p$ based on the elevations at three other points in the general vicinity. The coordinates and the water table elevations at these points are listed in Table 2.1. A structural analysis determined the semivariogram for the neighborhood of 20 $km$ to be linear with a slope of 4.0 $m^2/km$.

After solving the kriging equations to determine the weights, the estimate of the water elevation at point $p$ can be calculated. The kriging equations used to determine the weights are:

$$\begin{bmatrix} 0.0 & 1.0 & 1.0 & 1.0 \\ 1.0 & \gamma(h_{11}) & \gamma(h_{12}) & \gamma(h_{13}) \\ 1.0 & \gamma(h_{21}) & \gamma(h_{22}) & \gamma(h_{23}) \\ 1.0 & \gamma(h_{31}) & \gamma(h_{32}) & \gamma(h_{33}) \end{bmatrix} \cdot \begin{Bmatrix} \lambda \\ w_1 \\ w_2 \\ w_3 \end{Bmatrix} = \begin{Bmatrix} 1.0 \\ \gamma(h_{1p}) \\ \gamma(h_{2p}) \\ \gamma(h_{3p}) \end{Bmatrix}$$

Using the distance between the points, $h$, and the equation for the semivariogram, $\gamma(h_{ij}) = 4.0 * h$, the above equations are rewritten as:

$$\begin{bmatrix} 0.0 & 1.0 & 1.0 & 1.0 \\ 1.0 & 0.0 & 13.4 & 11.5 \\ 1.0 & 13.4 & 0.0 & 19.1 \\ 1.0 & 11.5 & 19.1 & 0.0 \end{bmatrix} \cdot \begin{Bmatrix} \lambda \\ w_1 \\ w_2 \\ w_3 \end{Bmatrix} = \begin{Bmatrix} 1.0 \\ 4.0 \\ 13.3 \\ 7.9 \end{Bmatrix}$$

Solving these equations produces the following estimates for the weights:

$$\begin{Bmatrix} \lambda \\ w_1 \\ w_2 \\ w_3 \end{Bmatrix} = \begin{Bmatrix} -0.7267 \\ 0.6039 \\ 0.0868 \\ 0.3093 \end{Bmatrix}$$

The elevation at $p$ is determined as:

$$X_0 = 0.6039 \cdot 120.0 + 0.0868 \cdot 103.0 + 0.3093 \cdot 142.0 = 125.3 \; meters$$

and the estimation variance is determined as:

$$Var(X_0 - X) = 0.6039 \cdot 4.0 + 0.0868 \cdot 13.3 + 0.3093 \cdot 7.9 - 0.7267 = 5.3 \; m^3/km$$

*Universal Kriging.* Many data sets are not stationary. There are two main causes of nonstationarity, global trend and local drift (6:238). As discussed in the structural analysis section, global trend applies to the entire data set and may be removed before using kriging. Local drift is trend in the neighborhood of the point being estimated. Therefore, a method of compensating for the local drift is needed. Universal kriging is used when local drift may be present. A nonstationary regionalized variable is composed of drift and the residual (7:393). The drift is the expected value of the variable in a neighborhood and the residual is the difference

between the drift and the actual value. In this form of kriging, the drift is removed from the regionalized variable by incorporating equations for the drift directly in the kriging equations, and the stationary residuals are kriged. In short,

> Universal kriging can thus be regarded as consisting of three operations: First, the drift must be estimated and removed. Then, the stationary residuals are kriged to obtain needed estimates. Finally, the estimated residuals are combined with the drift to obtain estimates of the actual surface. (7:393)

The drift is generally represented by a first or second-order polynomial. Davis (7:394–395) provides the matrix form of the universal kriging system when the first-order-polynomial drift at a point $p$ is defined as:

$$M_P = \alpha_1 X_{1i} + \alpha_2 X_{2i}$$

In this equation, the $\alpha$'s are drift coefficients which must be estimated and $X_{1i}$ and $X_{2i}$ are the coordinates of the $i^{th}$ control point.

The equations, in matrix form, are as follows:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 0 & X_1 & X_2 & \cdots & X_n \\ 0 & 0 & 0 & Y_1 & Y_2 & \cdots & Y_n \\ 1 & X_1 & Y_1 & \gamma(h_{11}) & \gamma(h_{12}) & \cdots & \gamma(h_{1n}) \\ 1 & X_2 & Y_2 & \gamma(h_{21}) & \gamma(h_{22}) & \cdots & \gamma(h_{2n}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_n & Y_n & \gamma(h_{n1}) & \gamma(h_{n2}) & \cdots & \gamma(h_{nn}) \end{bmatrix} \cdot \begin{Bmatrix} \lambda \\ \alpha_1 \\ \alpha_2 \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{Bmatrix} = \begin{Bmatrix} 1 \\ X_p \\ Y_p \\ \gamma(h_{1p}) \\ \gamma(h_{2p}) \\ \vdots \\ \gamma(h_{np}) \end{Bmatrix}$$

The corresponding minimized estimation variance of the estimate, $X_0$, is:

$$Var(X_0 - X) = \sum_{i=1}^{n} w_i \gamma(h_{ip}) + \lambda + \alpha_1 X_p + \alpha_2 Y_p$$

where $X_1, X_2, \ldots, X_n$ are the $x$ coordinates of the known points, $Y_1, Y_2, \ldots, Y_n$ are the $y$ coordinates of the known points, $X_p$ is the $x$ coordinate of the point being estimated, and $Y_p$ is the $y$ coordinate of the point being estimated. The covariance form of the equations and estimation variance are:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 0 & X_1 & X_2 & \cdots & X_n \\ 0 & 0 & 0 & Y_1 & Y_2 & \cdots & Y_n \\ 1 & X_1 & Y_1 & \sigma(h_{11}) & \sigma(h_{12}) & \cdots & \sigma(h_{1n}) \\ 1 & X_2 & Y_2 & \sigma(h_{21}) & \sigma(h_{22}) & \cdots & \sigma(h_{2n}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_n & Y_n & \sigma(h_{n1}) & \sigma(h_{n2}) & \cdots & \sigma(h_{nn}) \end{bmatrix} \cdot \begin{Bmatrix} \lambda \\ \alpha_1 \\ \alpha_2 \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{Bmatrix} = \begin{Bmatrix} 1 \\ X_p \\ Y_p \\ \sigma(h_{1p}) \\ \sigma(h_{2p}) \\ \vdots \\ \sigma(h_{np}) \end{Bmatrix}$$

and

$$Var(X_0 - X) = \sigma(0) - \left[ \sum_{i=1}^{n} w_i \gamma(h_{ip}) + \lambda + \alpha_1 X_p + \alpha_2 Y_p \right]$$

### *2.3 Data Set Minimization*

Data set minimization requires the repeated use of kriging to determine the minimal data set. The following procedure is used to determine a minimal data set (17):

1) Choose an initial set of points,

2) Calculate the unknown surface point values and estimation error,

3) Find the location and value of the largest estimation variance,

4) If the largest estimation error is greater than the maximum acceptable level of estimation error set by the user, add a point at that location and go to step 2, else stop the procedure.

In general, to produce an optimum minimal data set requires testing all possible combinations of the data and choosing the one with the fewest points that meets the maximum variance criteria. This would require great amounts of computational time even with a branch and bound algorithm. This method is described by Szidarovszky (18). The method chosen for this effort is, as explained previously, to add a point at the location of maximum estimation variance if that variance exceeds that required by the user. This will, in general, produce a suboptimal data set, but the difference between this and the optimal set will not be significant and is well worth the trade-off in computational time (17).

### *2.4 Summary*

Kriging and structural analysis topics from the literature were presented. This review presented the origin of kriging, defined kriging, and presented the kriging system of equations. Several of the more commonly used models for the theoretical semivariogram were discussed.

With the understanding of the kriging process and data set minimization provided by this literature review, the objective of this effort can be restated more precisely. Therefore, the objective of this effort is to develop a procedure that produces a minimal data set, in a reasonable amount of time, that meets the user provided maximum acceptable level of estimation error using the kriging process. The procedure must also perform surface kriging without minimization. The minimal data set produced is allowed to not be the optimal data set due to computational time limitations.

# *III. Methodology*

This chapter presents the methodology used in completing the objective outlined in Chapter I. Kriging involves both the structural analysis of the data and the determination of the estimates and error variances. These two activities were treated as separate tasks. The first task is discussed below under *Structural Analysis* and the second task is considered under the heading of *Kriging.* The techniques for minimization are discussed under the heading of *Minimization.* The fourth section documents the application of the kriging procedures.

*Supporting Tasks.* In support of the kriging procedures and the need for a common interface that isolates the input data format from the kriging procedures, two packages, written in C++, are provided. The first is a matrix object that provides for creation and manipulation of matrices. The second is an interface routine package that isolates each procedure from the data format and the control information input. These packages are documented in Appendix E, The Programmers Manual.

## *3.1 Structural Analysis*

Structural analysis is key to the efficient and optimal implementation of kriging in any field. This analysis must partition the data into homogenous groupings if necessary, ensure the global stationarity of the data by removing any global trend, calculate the experimental semivariogram, and estimate the parameters of the three commonly used theoretical semivariogram models.

*3.1.1 Trend Analysis* A value for the trend is needed at arbitrary locations. A method that provides exact values at those arbitrary locations is preferred over a method that requires estimating the values at those locations (17). Therefore, the

method of choice for removing global trend is to fit a polynomial of the form:

$$A + Bx + Cy + Dx^2 + Ey^2 + Fxy = z$$

through the data. A least squares regression technique is used to determine the coefficients. The program uses the common interface routines to get control information from the control file and to read in and write out the data. The trend may be removed from the data before or after it is partitioned. This program was written in C++ by Wayne McGee (14).

*3.1.2 Cluster Analysis* The following are the steps involved in partitioning the data into homogenous groupings. As the data is read in the row and column sums are calculated and the row and column sum medians are then estimated. The next step is to identify the rows and columns whose sum is above or below the corresponding median. The last step is to write the partitioned data to the disk along with information about the partitions. The data set may be partitioned before or after trend removal. This program was written in C++ by Donald Duckett (8).

*3.1.3 The Semivariogram* There are two activities involved with determining the semivariogram; calculating the experimental semivariogram and calculating the parameters of the theoretical semivariogram model. These two activities are addressed separately.

*Experimental Semivariogram.* The data may not benefit from partitioning and may not have a global trend. In that case, the only step to be performed on the data is the determination of the experimental semivariogram and fitting a model to that experimental semivariogram data.

The first task of this step is to determine the pairs of points that are a multiple of $\triangle h$ apart in the directions of interest to the user, usually $0^0$ (positive y direction) and $90^0$ (positive x direction). The determination of the experimental semivariogram

in each direction of interest is performed sequentially. Typically, data are known either at regularly spaced grid points or at irregularly distributed points throughout the region of study. Data is assumed to be available in the form $Z(x_i)$, where $Z(x_i)$ is the value of a random function at vector location $x_i$.

For points that are irregularly distributed throughout the region of study, the method of determining the pairs of points is more involved and requires more computation time. The angle and distance between each pair of points must be calculated. Those pairs of points that are in the direction of interest, $\alpha$, and at a multiple of $\Delta h$ apart are used in the determination of the experimental semivariogram. Since points may not be in exactly the direction $\alpha$, the user is allowed to specify a semi-inclusion angle, or regularization angle, $\phi$. If the angle between a pair of points, $Z(x_i)$ and $Z(x_j)$, falls within the range $\alpha \pm \phi$ then that point is considered to be in the direction of interest. If the pair of points are to be used in the calculation of the experimental semivariogram then the following two actions are performed: 1) the square of the difference of the quantity of interest of that pair is calculated, $(Z(x_i) - Z(x_j))^2$, and added to an array based on the distance between the points; and, 2) the location in the array containing the number of pairs of points integer multiples of $\Delta h$ apart is incremented based on the distance between the points. These two arrays are used later in the calculation of the experimental semivariogram.

The calculation of the experimental semivariogram is simpler when the data are aligned in a grid structure because the coordinates of each data point are a multiple of some incremental distance $\Delta h$ apart and the $0^0$ and $90^0$ directions are easily determined. The determination of the pairs of points $\Delta h$ apart in the $0^0$ and $90^0$ directions reduces to scanning the data in the y and x directions and calculating the square of the difference of the quantity of interest of that pair. The two summing arrays, mentioned previously, are updated for use later in the calculation of the experimental semivariogram.

The next step is to calculate the experimental semivariogram, described in Chapter II, using the two summing arrays and the formula:

$$\gamma^*(h) = \frac{1}{2|N|} \sum_{i=1}^{|N|} [z(x_i + h) - z(x_i)]^2$$

where $|N|$ is the number of pairs of data values at a distance of $h$ apart from one another, $x_i$ is the location of point $i$, $x_i + h$ is the location of a point at distance $h$ from $i$, and $z(x_i)$ and $z(x_i + h)$ are the values of the quantity of interest at $i$ and $x_i + h$.

The experimental semivariogram is a discrete function and can not be used at arbitrary locations in the data field. The theoretical variogram, however, is a continuous function. Therefore the last step is to calculate the required parameters for the theoretical semivariogram.

*Theoretical Semivariogram.* Three theoretical semivariograms are modeled using the experimental semivariogram calculated by the method outlined in the previous section. The three models are the linear, De Wijsian, and spherical models mentioned previously. The parameters of the theoretical semivariograms are estimated using a weighted least squares regression technique. The program to calculate both the experimental and theoretical semivariograms is written in C++ by Dr. David Robinson with modifications by Donald Duckett and Wayne McGee (17).

*3.1.4 Data Set Reconstruction* After calculating an estimate of the surface, based on the input data set, any removed trend must be added and any partitioned data must be reassembled.

*Trend Addition.* The global trend is added using a polynomial of the form:

$$A + Bx + Cy + Dx^2 + Ey^2 + Fxy = z$$

where the coefficients were previously calculated by the trend removal program. The program uses the common interface routines to get control information, including the polynomial coefficients, from the control file and to read in and write out the data. The trend must be added to the data before or after reassembly of the partitions based on when it was removed. This program is written in C++ by Wayne McGee (14).

*Partition Assembly.* If the data is partitioned it must then be manually reassembled. A program to do this was not written due to lack of time.

## *3.2 Kriging*

To perform the tasks required by this thesis effort and the two concurrent thesis efforts, a kriging program originally written in C by Michael Grant (9) was rewritten in C++ with extensive modifications. The modifications correct some programmatic errors as well as incorporate geometric anisotropy, minimal data set production, and generalization to any data set as required for satisfaction of the objective outlined in Chapter I. These routines are documented in Appendix E, The Programmers Manual. Note that a point is "kriged" if the estimate and the estimation variance at that point are calculated.

## *3.3 Minimization*

Minimization is the process of selecting a subset of the original data set subject to some constraint on the reconstruction process. For this effort that constraint is the maximum acceptable level of error set by the user.

*3.3.1 Background* The resolution is the number pixels (picture elements) that comprise the picture. More pixels means better resolution and as technology advances picture resolution increases. Mass production also allows the cost of the

higher resolution technology to decrease. These two facts have allowed picture producing devices to invade almost all aspects of every day life.

But these advances are not without their price. Higher resolution means increased information storage and transfer requirements. For example, a picture with a resolution of 1024 pixels by 1024 pixels with 3 bytes (24 bits) per pixel, 1 byte for each color component (red, green, and blue), needs 3 megabytes of information saved or transferred for each frame of that picture. Current telephone lines and equipment do not have sufficient bandwidth to transfer this amount of information in anything approaching real time as would be needed for teleconferencing. The standards they were based on did not envision or allow for video information transfer.

Storing these huge information files will cost a large amount of time and transferring then from place to place is difficult. The solution to both problems is some sort of data compression or minimization. Data compression techniques wreak great violence on the data by taking advantage of redundancies in the data and the nonlinear operation of the human eye. By exploiting correlation in space of still images and separate frames of video data, and correlation in time between frames for video data, large compression gains can be realized. For still images compression ratios of 10:1 to 50:1 (2) can be achieved. For video data compression ratios of 50:1 to 200:1 (2) can be achieved. But these methods are lossy in that the reconstructed images are quantitatively and qualitatively different than the original. This and other lossy techniques discard much of the original data and rely on the human eye to interpret the reconstructed image properly. This is acceptable in many applications but some, such as medical imaging, must have high quality reconstruction.

For those applications that require that there be little or no qualitative difference between the original and reconstructed image a different technique must be used. The compression ratio will be much lower for these data sets, as low as 2:1 for very high quality image reconstruction and as high as 4:1 for low quality image reconstruction. This research effort is aimed at satisfying the needs of those applica-

tions that require high quality image reconstruction. The following section describes several proposed methods of statistically minimizing this type of data.

*3.3.2 Szidarovszky's Methods* Szidarovszky describes a number of methods on how data set minimization can be accomplished (18:334-336) (3:193-195). Six of the methods is presented in this section.

*Method 1.* This method minimizes the estimation variance subject to a given number of additional measurement locations or cost (18:334). This model assumes there are $k$ existing known points and that $n-k$ additional points are to be added to the data set such that their choice of location minimizes the estimation variance. If there are no existing known points then $k = 0$. All remaining points are added to the data set and then these points are removed, one at a time, with the data set being tested after each removal and the point that yields the lowest estimation variance upon removal is discarded. This is repeated until $n$ points remain in the data set. The optimal selection method is based on assuming the following two monotonic properties of the estimation variance: 1) After increasing $n$ by one by adding the point $x_{n+1}$ then:

$$Var(n+1, x_{n+1}) \leq Var(n, x)$$

and 2) After decreasing $n$ by one by removing the point $x_n$ the estimation variance increases based on the previous observation. By use of enumeration in a tree search procedure the optimal data set can be selected. Therefore, this method starts by adding all the points to the data set and then removing them one by one until there are $n$ points remaining.

*Method 2.* This method minimizes the number of additional measurement locations or cost subject to upper bounds given to the estimation variances (18:336). This method is substantially the same as *Method* 1. The three main differences are: 1) Start with no additional points added; 2) add points instead of removing them; 3) Perform the test $M \leq \epsilon$ at each new node where $\epsilon$ is the upper bound on the

estimation variance and $M$ is the current estimation variance of the data set.

*Method 3.* This method determines the locations of a fixed number of measurement points using total enumeration and ects the set with the smallest estimation variance (3:193). The initial data set contains $k$ existing data points and $n-k$ additional points are to be added to this set from $N$ available points. Assuming $N > n-k$ the total data set to be tested for minimal estimation variance is given by:

$$\frac{N!}{(N-n+k)!(n-k)!}$$

Using a search tree, test for one of the following conditions at each node: 1) the number of elements of the subset of the data to be added to the data set equals $n-k$; or 2) all nodes which are endpoints of arcs starting from this node have all ready been searched. If either conditions holds then proceed backward from this node, otherwise proceed forward to the next point which has not been searched. Moving forward is equivalent to adding a point to the set while moving backward is equivalent to removing a point from the set. The initial subset, or node, has no points.

*Method 4.* This method determines the locations of a fixed number of measurement points using enumeration constrained by a branch and bound procedure and selects the set with the smallest estimation variance (3:193). This method is very similar to *Method* 3, but there are two major differences. The first difference is in the construction of the search tree. The initial node, or subset, contains all the points available to add to the data set and moving along each arc, from node to node, is equivalent to removing one point from the subset. The second difference is an additional condition that must also be checked: the current estimation variance is not less than the smallest one found for subsets containing exactly $n-k$ points. In this case removing more points makes the estimation variance larger than a solution previously determined.

*Method 5.* This method starts with an initial data set and sequentially includes additional measurements at the locations that yield the smallest estimation variance and terminates upon reaching the desired number of points (3:194). This method is similar to *Method* 3 in that the initial subset is empty but the selection of the next point to add to the subset is the one that produces the smallest estimation variance. This is continued until $n - k$ points have been added to the subset.

*Method 6.* This method starts with an initial data set containing the desired number of points and sequentially exchanges points from this data set with points not included this data set keeping the exchanges that yield the smallest estimation variance (3:195). For this method let $X_0 = (t_1, \ldots, t_{n-k})$ and $X_1 = (t_{n-k}, \ldots, t_N)$ and $j = 1$. Try to exchange the $j^{th}$ element of $X_0$ systematically with the elements of $X_1$. The exchange that minimizes the estimation variance is kept. If no exchange can decrease the estimation variance then do not make any exchanges and modify $j$ as follows:

$$j = \begin{cases} j+1 & \text{if } j < n-k \\ 1 & \text{if } j = n-k \end{cases}$$

and try exchanging the $j^{th}$ element of $X_0$ optimally. The procedure terminates when no exchange decreases the estimation variance.

*3.3.3 Brodkin's Method* The method chosen for implementation for this effort is a combination of *Method* 2 and *Method* 5 with slight modifications. This method minimizes the number of additional points to add to an initial data set subject to upper bounds given to the estimation variances. Additional points are sequentially included at the locations that yield the smallest estimation variance. Therefore, to find the minimal data set, the following steps must be performed:

*Step 1.* Compute the estimate and estimation variances for all points not included in the initial data set.

*Step 2.* Compare the largest estimation variance with the maximum allowed value specified by the user. If the calculated value is larger than the required maximum add a point at that location and go to step 1, otherwise stop.

*Gridded Data.* If the data is gridded, that is regularly spaced with the data grid fully populated, then the following technique is developed and presented as a method of selecting the minimal data set. The method takes maximum advantage of the gridded nature of the data to minimize the computational time. Also, if the correct initial pattern and pattern size is chosen, the minimal data set generated will be optimal or barely suboptimal.

As stated previously in the literature review, the estimation variance at a point is a function only of the relative distance and orientation of the known points used to estimate that point. This function is captured in the theoretical semivariogram model. Therefore, if a pattern of points could be chosen such that the largest estimation variance is just within the maximum allowed variance then that pattern of points would constitute the optimal minimal data set.

The method required for minimizing one dimensional gridded data will help illustrate the mechanics of this minimization method and is developed first. The method will then be extended to two dimensional gridded data. If all the points are aligned on an axis then the largest estimation variance would be at the midpoint of the two points which, without any intervening points, are farthest apart. This then reduces the problem to ensuring that the distance between any pair of consecutive points is just less than that which produces the maximum allowed variance. If the points are closer than this distance then the data set may not be minimal. If the points are farther apart than this distance then the maximum allowed variance criteria won't be met.

This analogy can be extended to multiple dimensions with little difficulty. For this effort, a two dimensional pattern is desired such that the maximum allowed

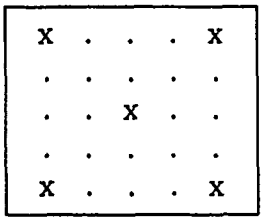


Figure 3.1. Basic Kriging Pattern

variance criteria is met. Kriging does not produce good results when a point beyond an edge of the data field is estimated. Therefore, the corner points of the area of interest must be included in the minimal data set. This suggests a square or rectangular pattern but it was felt that including the center point, producing an "X" pattern, would produce better results by needing less data set for the same maximum allowed variance. Therefore, the "X" pattern based on the semi-inclusion distances shown in Figure 3.1 is typical. Where "x" is a known point and "." represents points to be estimated.

At this point the semi-inclusion distance must be defined. The distance between the center point and a corner point must be less than the semivariogram range, $a$. This allows any unknown point within the square to be estimated based on no fewer than two known points, the center point and one or more corner points. The $x$ semi-inclusion distance, IAX, is then defined as the number of rows between the center point and a corner point plus one. The $y$ semi-inclusion distance, IAY, is then defined as the number of columns between the center point and a corner point plus one. The distance between the center point and a corner point must not be larger than the semivariogram range, but, it may be substantially smaller.

If the inclusion distance, twice the semi-inclusion distance, is too large, then the largest estimation variance will be larger than the maximum allowed variance. Therefore, a point will need to be added to the data set. This first point will be added at the midpoint of the longest edge and, due to symmetry, a point will also

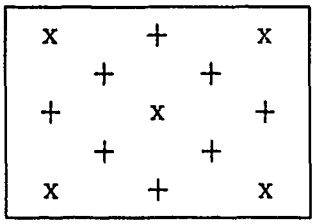


Figure 3.2. Replicated Basic Kriging Pattern

be added to the other longest edge. If the maximum variance criteria is still not met then two more points, one at the midpoint of each remaining edge, will be added. If the maximum variance criteria is still not met then four points, due to symmetry, will be added, one at each of the midpoints of the lines joining the center point and each corner point. This will result in the pattern shown in Figure 3.2. Where "+" represents an added point and unknown points are omitted for clarity. It can be seen that the original pattern is now replicated four times within the original data field. If the semi-inclusion distance is divisible by two then the replicated pattern will be identical to the original pattern except for scale. If the pattern is exactly replicated the size of the problem can be reduced by a factor of four, that is a factor of two in each direction. This will greatly reduce the computational time. It should also be noticed that the weights calculated for one of the replicated patterns are identical to the weights calculated for the other replicated patterns. These two facts are used to great advantage in the minimization programming.

In order to ensure that the estimation variance at a point being estimated is minimized, surrounding points must be included in the kriging pattern if possible. Therefore, the points that are twice the semi-inclusion distances in each direction from the center of the pattern are included as part of the kriging pattern for areas located in the central region of the data field. This results in the pattern shown in Figure 3.3. Where "x" is a known point within the area being estimated and "s" is a surrounding point included to minimize the estimation variance. Only the points

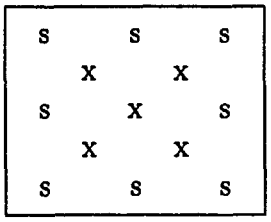



Figure 3.3. Augmented Kriging Pattern

| | | | | |
|---|---|---|---|---|
| 6 | 3 | 3 | 3 | 7 |
| 2 | 1 | 1 | 1 | 4 |
| 2 | 1 | 1 | 1 | 4 |
| 2 | 1 | 1 | 1 | 4 |
| 8 | 5 | 5 | 5 | 9 |

Figure 3.4. Kriging Areas

bounded by lines connecting the corner "x" points are being estimated. The edges and corners must be estimated separately from the central area because there are no points beyond the edges to include in the pattern and the point pattern on the edges and corners will not be identical to the central area point patterns. If the point patterns are not identical then neither are the weights. This creates nine separate areas to krige as shown in Figure 3.4. As the figure illustrates, the nine areas are the four corners, the four edges, and the central section. For clarity it is not shown, but the edges of each kriging area overlaps the edges of two or more other kriging areas. The weights only need to be calculated once for one kriging area for the edge and central sections and can be used for each identically patterned area in that section. This will mean a great savings in computational time.

By now it should be apparent that obtaining the optimal minimal data set is heavily dependent on the initial values of the semi-inclusion distances. If these values are not an integer multiple of the ideal values then too many points will be added to meet the maximum allowed variance criteria. Therefore, an iterative routine was

written to determine these ideal values. All the points within a kriging area need not be estimated to determine the largest estimation variance. Only those points within a small area located at the maximum distances from the known points need to be estimated to determine the maximum estimation variance within that kriging area. This method for determining the ideal semi-inclusion distances also produces a large savings in computational because fewer points are added and hence fewer points need to be estimated again based on the added points. These routines are documented in Appendix E, The Programmers Manual.

It has not been explicitly stated but the matrix of semivariogram values includes all the points in the kriging pattern, even those that are outside the theoretical semivariogram range of the point being estimated. If this were not so then a new $[A]$ matrix would be required for every point to be estimated and the great computational time savings would be lost. Also not explicitly stated is the requirement to add points to the entire grid such that the current kriging area pattern is replicated throughout the grid.

Of interest (for the pattern minimization method) is the implementation of a matrix inversion method that takes advantage of the prior matrix inversion. This is of interest due to the fact that adding points to the pattern adds rows and columns to the $[A]$ matrix and points are added based on estimation variances calculated by inverting the prior $[A]$ matrix. This routine uses a matrix partitioning method to reinvert the matrix (19:192). Given the two matrices, $[A]$ and $[B]$, partitioned into submatrices as follows:

$$\left[ A \right] = \begin{bmatrix} P & Q \\ R & S \end{bmatrix} \quad \left[ B \right] = \begin{bmatrix} Y & Z \\ U & V \end{bmatrix}$$

where $[A] \cdot [B] = [I]$ and $P^{-1}$ is known. Performing the multiplication it can be seen that:

$$P \cdot Y + Q \cdot U = I \tag{3.1}$$

$$P \cdot Z + Q \cdot V = 0 \tag{3.2}$$

$$R \cdot Y + S \cdot U = 0 \tag{3.3}$$

$$R \cdot Z + S \cdot V = I \tag{3.4}$$

solving the second equation for $Z$:

$$Z = -P^{-1} \cdot Q \cdot V$$

combining this with the fourth equation and solving for $V$:

$$V = (S - R \cdot P^{-1} \cdot Q)^{-1}$$

solving the first equation for $Y$:

$$Y = P^{-1} \cdot (I - Q \cdot U) = P^{-1} - (P^{-1} \cdot Q) \cdot U$$

combining this with the third equation yields:

$$U = -(S - R \cdot P^{-1} \cdot Q)^{-1} \cdot R \cdot P^{-1} = -V \cdot R \cdot P^{-1}$$

therefore, the following relationships are established:

$$\begin{aligned} V &= (S - R \cdot P^{-1} \cdot Q)^{-1} \\ Z &= -P^{-1} \cdot Q \cdot V \\ U &= -V \cdot R \cdot P^{-1} \\ Y &= P^{-1} - (P^{-1} \cdot Q) \cdot U \end{aligned}$$

If the number of added rows and columns is small relative to the original matrix size, the computational time saved by not inverting the entire matrix is substantial.

### *3.4 Procedure Application*

The data used for proof of concept are Magnetic Resonance Image (MRI) brain scans. The scans are comprised of several parallel planes of data in which each plane, or slice, is 268 measurements, (pixels) wide and 267 measurements, (pixels) long. Each measurement is an intensity represented as a grey scale value between 0 and 255. The values can be represented by 1 byte unsigned integers. Therefore, one brain scan, containing 60 slices, will need 4 megabytes of disk space. This type of data is used for proof of concept due to the requirement that the quality of the reconstructed image be as good as the original image. Since kriging is not a lossy technique, this type of data is ideally suited for kriging minimization. For this application the pixel location in each slice, row and column, are used as the coordinates of the point and the grey scale value is used as the quantity of interest at each point.

Using the procedures developed in the previous sections, minimal data sets for various maximum variances and initial data patterns are found. The following summarizes the steps taken to produce the minimal data sets.

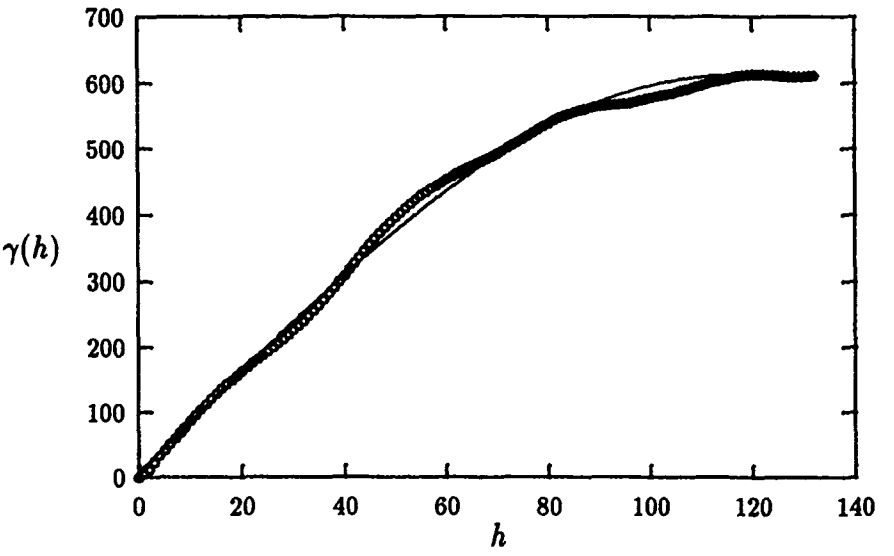
*Step 1. Data Partitioning.* Only slice 24 is partitioned for proof of concept. The partitioning of slice 24 is illustrated in Figure 3.5. Not all the data sets would benefit from partitioning. The partitions that the data set may be divided into may be too small to appreciably decrease the number of points in the minimal data set and is, therefore, not worth the extra effort involved in partitioning and reassembling the data set. Slice 1 is a good example of a data set that would not benefit from partitioning. The central image would be in one partition and the background would be in four very thin edge partitions and four very small corner partitions. This partitioning may actually increase the number of points in the minimal data set since the corner points of each partition must be included in the minimal data set

Figure 3.5. Partitioning of Slice 24

so that points beyond the edge of the data set aren't estimated.

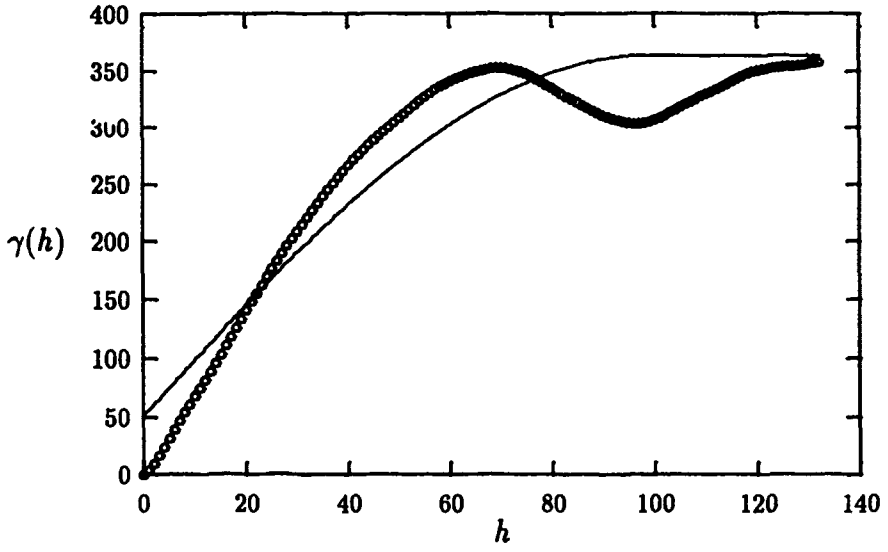
*Step 2. Trend Removal.* Global trend is not removed from any of the data sets (17).

*Step 3. Semivariogram Determination.* The data showed both geometric and zonal anisotropy as illustrated in Figure 3.6 and Figure 3.7. The kriging program can accommodate the different ranges due to geometric anisotropy and averages the sills to remove the zonal anisotropy from the theoretical semivariogram model (not from the data). The theoretical spherical semivariogram model is plotted with the experimental semivariogram in Appendix B for the $0^0$ and $90^0$ directions of every slice of MRI data used.




$$\gamma(h) = 602.753\left[\left(\frac{3}{2}\right)\left(\frac{h}{114.949}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{114.949}\right)^3\right] + 8.171$$

Figure 3.6. Slice 12 Semivariogram in the $0^0$ Direction




$$\gamma(h) = 311.715\left[\left(\frac{3}{2}\right)\left(\frac{h}{97.840}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{97.840}\right)^3\right] + 51.756$$

Figure 3.7. Slice 12 Semivariogram in the $90^0$ Direction

*Step 4. Kriging of Data Sets.* Using the kriging program, minimal data sets for three maximum allowed variance levels and five initial data patterns were produced. The three maximum allowed variances were chosen such that the same values could be used for all slices without causing all the known points to be included in the minimal data set. A second requirement was that the values chosen would produce recognizable images of slice one with a minimum difference between the values of fifteen. This was accomplished by iteratively kriging all the slices using integer values for the maximum allowed variances until the desired results were obtained. The five semi-inclusion distances were chosen by requiring a square pattern and counting up from one. Therefore, the three maximum allowed variances were 100.0, 85.0, and 70.0. The five initial data patterns were for the semi-inclusion distances of 1, 2, 3, 4, and 5. The partitioned slice was minimized with maximum allowed variances of 11.0, 12.0, and 13.0. These values were chosen such that the image quality produced was comparable to that obtained using semi-inclusion distances of 1, 2, and 3. This was also accomplished by iteratively kriging the data set. A visual inspection was performed to determine performance of the procedure. A display program was used to display the slices for visual inspection. Photographs of the original and estimated slices are in Appendix C. The reduction percentage and largest variances of the first slice are shown in Table 3.1. Tables showing the percent reduction of data points, the largest variance, and the photograph number of the results of all the kriged minimal data sets are in Appendix D.

*Step 5. Trend Addition.* Because trend is not removed this step is not performed for these data sets.

*Step 6. Partition Assembly.* The one partitioned slice is reassembled before the visual inspection.

The steps summarized above are developed in the previous sections and demonstrate the process used in applying kriging in the production of minimal data sets. These steps provide the methodology for obtaining the minimal data sets.

Table 3.1. Slice 1 Statistics

| Minimization Method | | Largest Variance | Percent Reduction |
|---|---|---|---|
| Maximum Allowed Variance | 100 | 98.98 | 98.70 |
| | 85 | 84.39 | 96.53 |
| | 70 | 68.37 | 74.90 |
| Initial Semi-Inclusion Distances IAX, IAY | 5, 5 | 91.56 | 97.88 |
| | 4, 4 | 85.72 | 96.77 |
| | 3, 3 | 79.72 | 94.34 |
| | 2, 2 | 73.50 | 87.36 |
| | 1, 1 | 66.92 | 50.00 |

As this chapter has shown, the procedures developed can produce a minimal data set in a statistically controlled manner. The controlling parameter is the maximum allowed variance set by the user. To aid the kriging program in obtaining the best results a trend removal program and a partitioning program were developed. To capture the structure of the data set in continuous function, a theoretical semivariogram model producing program was developed. To replace the removed global trend a rebuilding program was developed.

# *IV. Results and Conclusions*

This chapter includes the results of the research and several conclusions based on these results. As previously stated, the purpose of this research effort was to develop a procedure that could select a minimal data set in a statistically controlled fashion and estimate unknown values, at regular grid locations, based on a user desired overall maximum level of error. This goal was achieved. The following results and conclusions are provided with reference to the objective outlined in Chapter I.

## *4.1 Results*

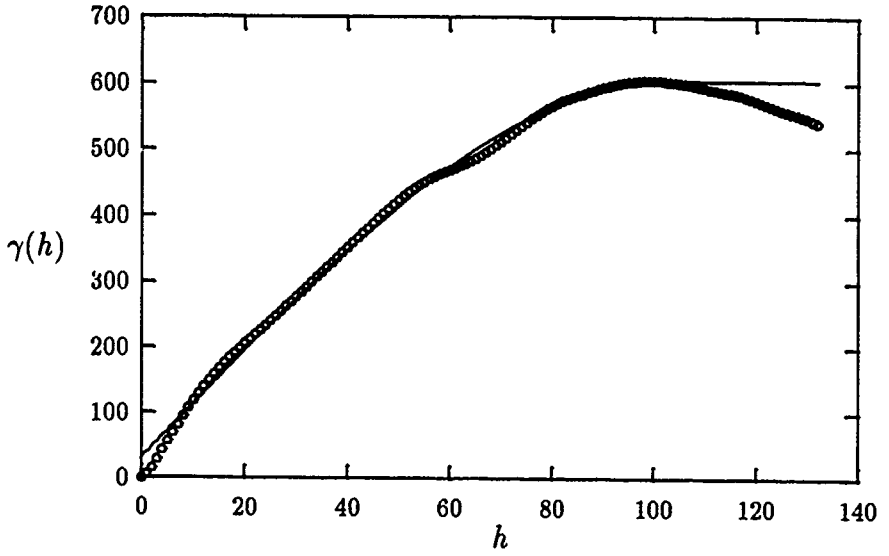
In general, the results of this effort are the procedures developed to produce minimal data sets. The procedures developed in Chapter III and the computer programs written for this effort provide the means for producing minimal data sets. Nine slices of Magnetic Resonance Image data of a baby's brain were used to demonstrate use of the procedures. The slices used are numbered 1, 3, 6, 9, 12, 15, 18, 21, and 24. These provided a representative cross section of the entire data set.

As mentioned previously, no global trend was removed. Five minimal data sets were produced for each slice using two different methods. The first method produced three minimal data sets based on maximum allowed variances of 100.0, 85.0, and 70.0 gray scale values squared. The second method produced five minimal data sets based on semi-inclusion distances for (IAX,IAY) of (5,5), (4,4), (3,3), (2,2), and (1,1). The three maximum allowed variances were chosen such that the same values could be used for all slices without causing all the known points to be included in the minimal data set. A second requirement was that the values chosen would produce recognizable images of slice one with a minimum difference between the values of fifteen gray scale values squared. This was accomplished by iteratively kriging all the slices using integer values for the maximum allowed variances until the desired results were obtained. The five semi-inclusion distances were chosen by

requiring a square pattern and counting up from one. For the data sets produced using the initial point pattern based on the semi-inclusion distances the maximum allowed variance was set to a value large enough to ensure that no points would be added to the data set. Slice 24 was also partitioned into a background and a center data partition and three minimal data sets were produced using maximum allowed variances of 10.0, 12.0, and 11.0 gray scale values squared. These values were chosen such that the image quality produced was comparable to that obtained using semi-inclusion distances of 1, 2, and 3. This was also accomplished by iteratively kriging the data set.

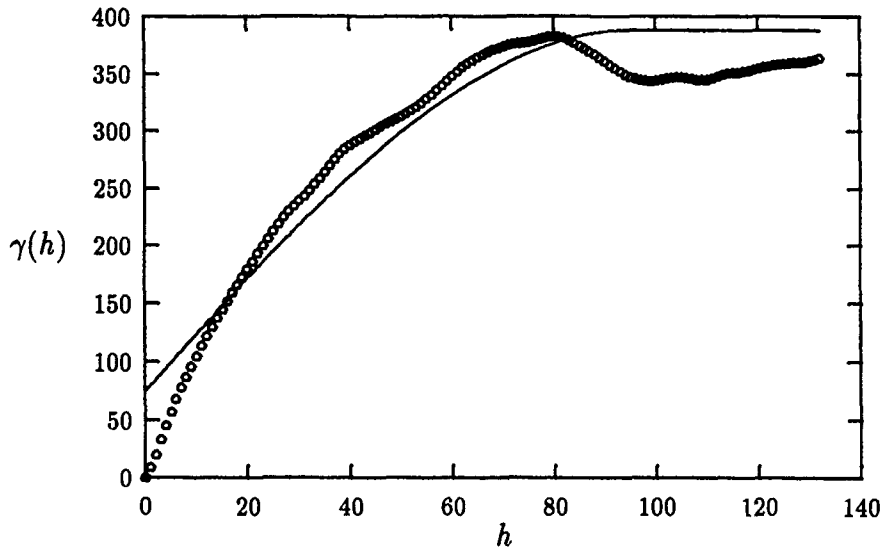
For this effort, the background of each image, which is normally filtered out by the display program, was considered to be known data along with the brain scan data.

Figure 4.1 and Figure 4.1 are plots of the experimental and theoretical semivariograms for slice one for the $0^0$ and $90^0$ directions and are typical for all the slices. The data has both geometric and zonal anisotropy. The plots of the experimental and theoretical spherical semivariograms for the $0^0$ and $90^0$ directions for all slices, including the partitioned slice, are in Appendix B.




$$\gamma(h) = 571.150\left[\left(\frac{3}{2}\right)\left(\frac{h}{102.994}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{102.994}\right)^3\right] + 32.231$$

Figure 4.1. Slice 1 Semivariogram in the $0^0$ Direction




$$\gamma(h) = 312.925\left[\left(\frac{3}{2}\right)\left(\frac{h}{95.117}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{95.117}\right)^3\right] + 75.038$$

Figure 4.2. Slice 1 Semivariogram in the $90^0$ Direction

As evidenced by Figure 4.3 and Figure 4.4, minimizing each slice based on the same maximum allowed variance did not produce the same quality of image for all slices. For the same maximum allowed variance of 70.0, slice 9 looks like a brain scan whereas slice 24 looks like an indistinct blob. However, Figure 4.5 and Figure 4.6 show that minimizing based on initial semi-inclusion distances did produce comparable quality images for all slices. This is due to the fact that the estimation variance is directly proportional to the sill, $C + C0$, minus the nugget effect, $C0$, of the theoretical semivariogram model. Therefore, the slices with smaller sill minus nugget effect values will have fewer points in the minimal data set for the same maximum allowed variance. With fewer known points from which to estimate the surface, the estimated surface is noticeably different than the original image. By using the semi-inclusion distances to determine the minimal data set, the same number of points was used to estimate the surface for each slice. Thus the difference between the original image and the estimated surface was much less noticeable for these minimal data sets. Therefore, by using semi-inclusion distances of (1,1), half the data may be removed with barely noticeable differences between the original and estimated images. By using partitioning on slice 24, as shown in Figure 4.7, a further reduction of 16.14 percent in the number of points was achieved with out a reduction in the quality of the image. The partitioned image is shown in Figure 4.8. Photographs of the results of the kriging procedures for all slices are in Appendix C. Each series of images has the original data set on the left, the kriged image in the center, and the error between the original and kriged image on the right.

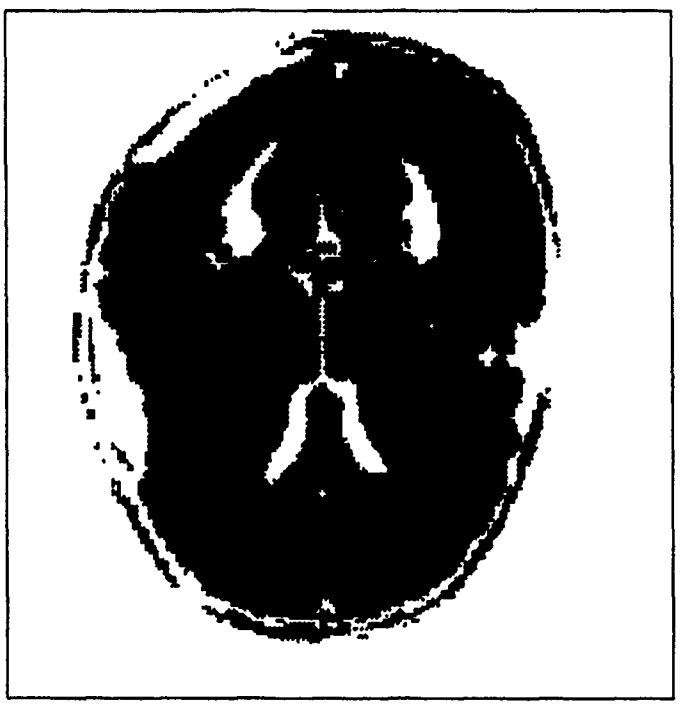


Figure 4.3. Slice 9, Maximum Variance = 70

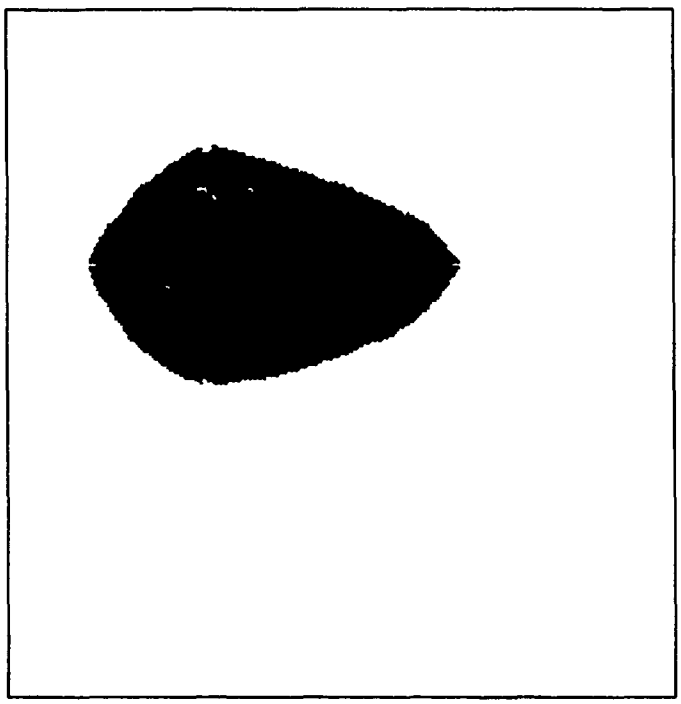


Figure 4.4. Slice 24, Maximum Variance = 70

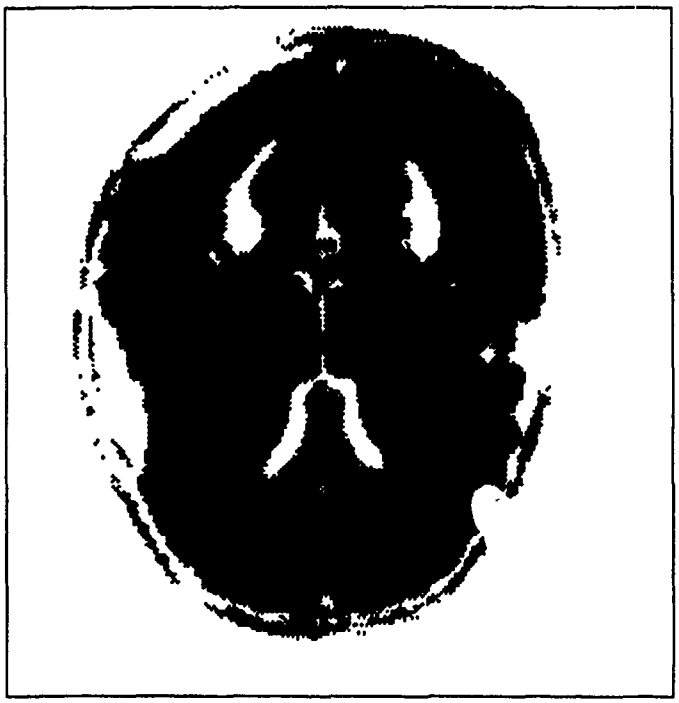


Figure 4.5. Slice 9, Semi-Inclusion Distances = (1,1)

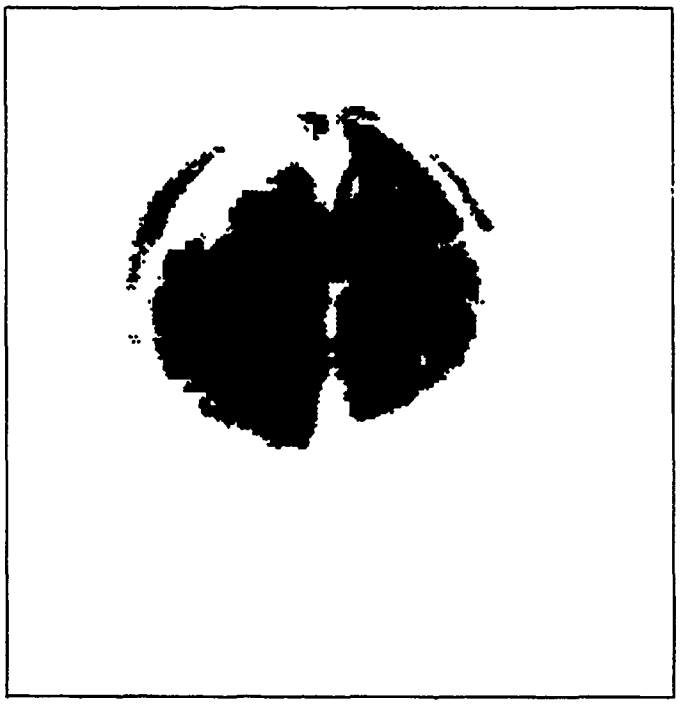


Figure 4.6. Slice 24, Semi-Inclusion Distances = (1,1)

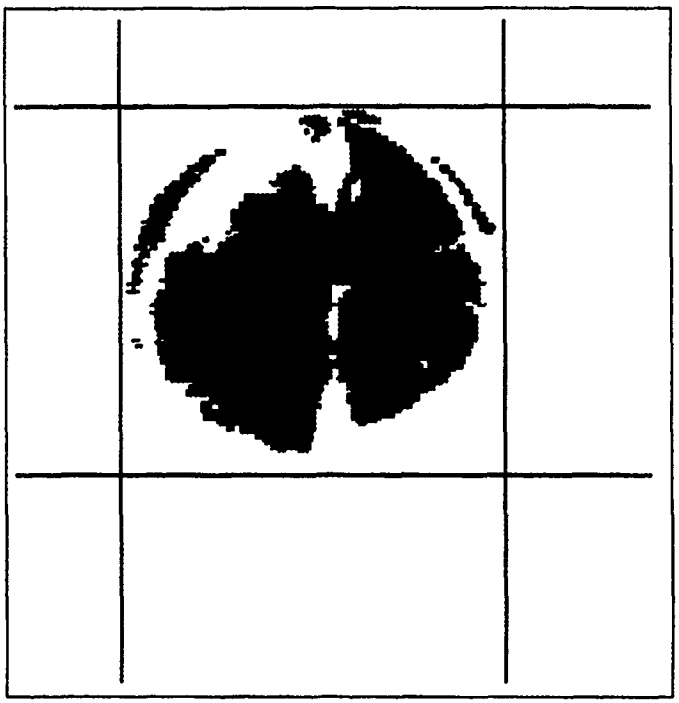


Figure 4.7. Slice 24 Partitioning

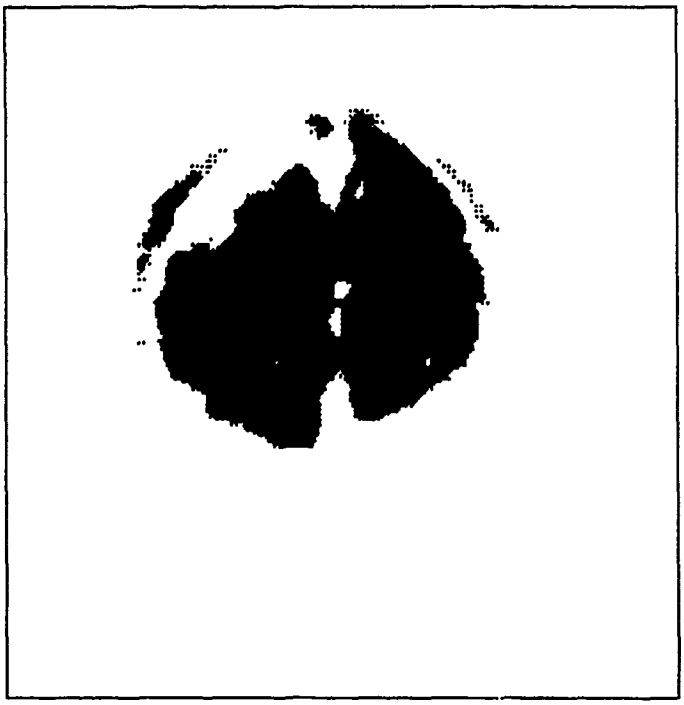


Figure 4.8. Slice 24 Partitioned Image

Table 4.1 and Table 4.2 show the statistics for slice 9 and slice 24. The magnitudes of the numbers in the tables are typical for all slices. The typical run times, for the minimal data sets produced using the kriging pattern, were under ten minutes. The run times for the minimal data sets that were not able to take advantage of the kriging pattern were as high as two hours. The statistics for all slices are tabulated in Appendix D.

Table 4.1. Slice 9 Statistics

| Minimization Method | | Largest Variance | Percent Reduction | Run Time hr:min:s | Photograph |
|---|---|---|---|---|---|
| Maximum Allowed Variance | 100 | 98.69 | 98.33 | 00:45:04 | 10(a) |
| | 85 | 82.09 | 93.63 | 00:39:55 | 10(b) |
| | 70 | 69.18 | 25.09 | 01:34:58 | 10(c) |
| Initial Semi-Inclusion Distances IAX, IAY | 5, 5 | 95.44 | 97.88 | 00:03:33 | 11(a) |
| | 4, 4 | 89.79 | 96.77 | 00:03:46 | 11(b) |
| | 3, 3 | 83.99 | 94.34 | 00:04:42 | 11(c),12(a) |
| | 2, 2 | 77.97 | 87.36 | 00:05:51 | 12(b) |
| | 1, 1 | 71.61 | 50.00 | 00:19:48 | 12(c) |

Table 4.2. Slice 24 Statistics

| Minimization Method | | Largest Variance | Percent Reduction | Run Time hr:min:s | Photograph |
|---|---|---|---|---|---|
| Maximum Allowed Variance | 100 | 53.15 | 99.96 | 01:42:16 | 25(a) |
| | 85 | 53.15 | 99.96 | 02:02:15 | 25(b) |
| | 70 | 53.15 | 99.96 | 01:41:20 | 25(c) |
| Initial Semi-Inclusion Distances IAX, IAY | 5, 5 | 3.94 | 97.88 | 00:04:06 | 26(a) |
| | 4, 4 | 3.15 | 96.77 | 00:04:25 | 26(b) |
| | 3, 3 | 2.36 | 94.34 | 00:05:18 | 26(c),27(a) |
| | 2, 2 | 1.57 | 87.36 | 00:07:36 | 27(b) |
| | 1, 1 | 0.79 | 50.00 | 00:20:29 | 27(c) |

Not shown in the appendices is the high degree of automation achieved. All the figures, tables, and photographs can be recreated in under two days using one Silicon Graphics 220 work station. This was achieved by generating all the required control files and running the programs in batch mode using a shell script.

The initial minimal data set points produced indicated that a square or rectangular pattern is better than an "X" pattern for producing the optimal minimal data set. This fact was used to develop the *Calculate_IAX_IAY* routine to determine the ideal initial semi-inclusion distances. The data sets were then kriged again to produce the optimal minimal data sets presented in this effort. Inspection of the tables in Appendix D reveals that, except for slice 24, the largest estimation variance of all the minimal data sets produced was within 3.2 percent of the maximum allowed variance. Due to the very low sill minus nugget effect of the theoretical semivariogram model for slice 24, which is directly related to the calculated estimation variance, the largest distance allowed for the semi-inclusion distances was not large enough to bring the calculated estimation variance up to the maximum allowed variance. The limit on the semi-inclusion distances allows all points that are to be estimated to be within range of two or more known points.

### *4.2 Conclusions*

In conclusion, this thesis develops and demonstrates the application of kriging in the controlled minimization of large data sets. Specifically, the following were shown: 1) A minimal data set can be selected based on a maximum acceptable level of error; 2) Minimal data sets based on the semi-inclusion distances produced more uniform quality in the reconstructed images for each slice; 3) For gridded data a rectangle pattern of known points produces a minimal data set that is closer to the optimal minimal data set than an "X" pattern of known points; 4) For large data sets (more than 71,000 points) containing gridded data the run times are very reasonable (typically under ten minutes); 5) Performing trend removal and partitioning on the

data can substantially improve the results; 6) Minimal data set selection can be fully automated; and 7) This procedure can be applied to any type of data.

In achieving the goal of this thesis, two objectives were accomplished. First, a viable kriging procedure was developed. This kriging procedure included the structural analysis of the data and the development of a universal kriging program for estimating the surfaces and the variances. Secondly, the procedures were demonstrated using Magnetic Resonance Image data and for this data a 2:1 compression ratio produced barely noticeable differences in the reconstructed image over the original image. Further research in this area needed, especially in the application of these procedures to three dimensional data.

# V. Recommendations

This chapter provides recommendations which suggest either improvements in this effort or areas for further research related to this study. As this thesis effort may very well be the first automated application of kriging in the controlled minimization of large data sets, further research in this area may prove promising. Recommendations are provided for all areas of this study and are presented for consideration.

## 5.1 Structural Analysis

A study concerning the sensitivity of the kriging procedures to the theoretical nugget effect should determine the robustness of the variogram structure.

A more sophisticated partitioning method would produce better results in images in which the anomalous feature is not aligned with the axis of the image.

## 5.2 Kriging

Extending kriging to three dimensions should be fairly straight forward and of great interest and benefit to many people. By using known points in adjacent slices then estimation variance, and thus the number of points in the minimal data set, could be further minimized. This technique could also be applied to other types of three dimensional data.

Since kriging involves using known points and not previously kriged points the procedure can be highly vectorized. Therefore, running the program on a vectorizing machine may provide real time kriging.

Allowing area specific semivariograms may produce tremendous improvements in image quality without resorting to partitioning.

### *5.3 Minimization*

Coding should be added to the kriging procedure that would add adjacent points to the minimal data set if the change in the quantity of interest is above a threshold value set by the user. This would improve the image quality while reducing the computational time otherwise required to add all those required points.

Coding should be added to properly minimize irregularly distributed data. Some method of finding the point nearest to the desired location of point addition needs to be added to the *Add_Pts* routine for non-gridded data.

The effects of calculating the true variance of the estimate instead of the theoretical variance may provide better results in image quality for the same maximum variance (at the cost of computational time since the pattern may be lost). This may produce minimal data sets with the same image quality give the same maximum allowed variance.

# *Bibliography*


1. Anderberg, Michael R. *Cluster Analysis for Applications*. New York: New York Academic Press, 1973.
2. Ang, Peng H. and others. "Video Compression Makes Big Gains," *IEEE Spectrum*, *28*(10):16–19 (October 1991).
3. Carrera, Jesus and Ferenc Szidarovszky. "Numerical Comparison of Network Design Algorithms for Regionalized Variables," *Applied Mathematics and Computation*, *16*:189–202 (1985).
4. Clark, Isobel. *Practical Geostatistics*. London: Applied Science Publishers Ltd, 1979.
5. Cressie, Noel. "Geostatistics," *The American Statistician*, *43*(4):197–202 (November 1989).
6. David, Michael. *Geostatistical Ore Reserve Estimation*. New York: Eisevier Scientific Publishing Company, 1977.
7. Davis, John C. *Statistics and Data Analysis in Geology* (Second Edition). New York: John Wesley & Sons, 1986.
8. Duckett, Donald P. *The Application of Statistical Estimation Techniques to Terrain Modeling*. Draft MS thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1991.
9. Grant, Michael. *The Application of Kriging in the Statistical Analysis of Anthropometric Data Volume I*. MS thesis, AFIT/GOR/ENY/ENS/90M-8, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, May 1990 (AD-A220 613).
10. Henley, Stephen. *Nonparametric Geostatistics*. New York: John Wiley & Sons, 1981.
11. Journel, A. G. "Kriging in Terms of Projections," *Mathematical Geology*, *9*(6):563–86 (1977).
12. Journel, A. G. and Ch. J. Huijbregts. *Mining Geostatistics*. New York: Academic Press, 1978.
13. Matheron, G. "Principles of Geostatistics." In John M. Cubbit and Stephen Henley, editor, *Statistical Analysis in Geology*, Stroudsburg: Dowden, Hutchinson, & Ross, Inc., 1978.
14. McGee, Donald W. *The Application of Statistical Kriging to Improve Satellite Imagery Resolution*. Draft MS thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1991.

15. Omre, Henninh. "The Variogram and Its Estimation." In G. Verly, M. David, A. G. Journel, and A. Marechal, editor, *Goestatistics for Natural Resources Characterization*, Boston: D. Reidel Publishing Company, 1984.

16. Press, William H. and others. *Numerical Recipes in C*. Cambridge: Cambridge University Press, 1988.

17. Robinson, D. Class notes from kriging seminar. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 1991.

18. Szidarovszky, Ferenc. "Multiobjective Observation Network Design for Regionalized Variables," *International Journal of Mining Engineering*, *1*:331–42 (1983).

19. Szidarovszky, Ferenc and Sidney Yakowitz. *Principles and Procedures of Numerical Analysis*. New York: Plenum Press, 1978.

20. Yakowitz, S. J. and F. Szidarovszky. "A Comparison of Kriging with Nonparametric Regression Methods," *Journal of Multivariate Analysis*, *16*:21–53 (1985).

## *Vita*

Captain Christopher Brodkin was born on 23 February 1958 in Upland, California. He graduated from Alta Loma High School in Alta Loma, California in 1976. After high school, he joined the California Air National Guard as an aircraft crew chief. While in the Air National Guard, he attended the California State Polytechnic University in Pomona, California where he majored in Aerospace Engineering. He completed his degree in 1982 and was assigned as a Computer Systems Manager for the Air Force Weapons Laboratory, Technology Assessment Division at Kirtland AFB in Albuquerque, New Mexico. In November 1986, Captain Brodkin was reassigned to the Foreign Technology Division, Wright Patterson AFB, in Fairborn Ohio where he served as a foreign space launch vehicle assessment expert. He entered the School of Engineering at the Air Force Institute of Technology in August 1987 as a part time student and in July 1990 became a full time student. He earned a Master of Science Degree in Astronautical Engineering and a Master of Science Degree in Computer Engineering concurrently.

Captain Brodkin has two sons; Christopher Donavan, Junior, and William Evan.

Permanent address: 3053 Osage Street
Camden, Arkansas 71701

# Appendix A. *Users Manual*

This appendix is in three sections. The first section provides the program control variable names, usage, and programs actions based on those inputs. The second section contains two sample *Krige* control files. The third section contains a sample *Varfit* control file. To run the programs type:

*Program_Name Control_File*

where *Program_Name* is *Krige*, *Varfit*, *Rebuild*, *Partition*, or *Residuals* and *Control_File* is the name of the file that contains the program control variable names and their associated value. The interface routines may be modified to procur the program control variables from anywhere deemed more convenient by the user (8). The programs currently reside on *Poincare*, a Silicon Graphics 220 work station, under the directory */d2/cbrodkin/krige_code*.

## *A.1 The Inputs*

The input names are case insensitive. If an input is specific to one program then the program to which that input is specific is given. The "=" is part of the variable input name and there can not be any embedded blanks in the input name. The "-" is a separator between the variable input name and its description. Text in the control file not recognized as a program control variable name is ignored. Therefore, as shown in the sample *Krige* control files, comments describing the input may be in the control file.

*End_Of_Input_Values* - Signals end of input. Can be used in data files containing input in a header section. The version of the interface used for this thesis effort does not allow a data file header to contain program input.

*Data_Filename=* - Specifies the name of the file containing the input data. Can not be longer than seventy characters. If not provided the programs will print an error message and terminate.

*Output_Filename=* - Specifies the name of the file in which to write the output data. Can not be longer than seventy characters. If not provided then no output to this file will be written.

*Variance_Filename=* - Specifies the name of the file in which to write the estimation variance data. Only applies to the *Krige* program. Can not be longer than seventy characters. If not provided then no output to this file will be written.

*Minimal_Dataset_Filename=* - Specifies the name of the file in which to write the minimal data set. Only applies to the *Krige* program doing minimization. Can not be longer than seventy characters. If not provided then no output to this file will be written.

*Plot_Filename=* - Specifies the base name of the files in which to write the experimental semivariogram data in two column format suitable for plotting. A "0" is postpended for the file containing the $0^0$ direction experimental semivariogram

data. A "90" is postpended for the file containing the $90^0$ direction experimental semivariogram data. Only applies to the *Varfit* program. Can not be longer than seventy characters. If not provided then no output to these files will be written.

*Variogram_Filename=* - Specifies the name of the file in which to write the experimental semivariogram data and the parameters of the spherical, linear, and dewijsian theoretical semivariogram models. Only applies to the *Varfit* program. Can not be longer than seventy characters. If not provided then no output to this file will be written.

*Error_Filename=* - Specifies the name of the file in which to write any program error messages. Can not be longer than seventy characters. If not provided then output will be directed to the screen.

*A(0)=* - Specifies the theoretical semivariogram range in the $0^0$ direction.

*A(90)=* - Specifies the theoretical semivariogram range in the $90^0$ direction.

*C(0)=* - Specifies the spherical model theoretical semivariogram sill minus nugget effect in the $0^0$ direction.

*C(90)=* - Specifies the spherical model theoretical semivariogram sill minus nugget effect in the $90^0$ direction.

*C0(0)=* - Specifies the spherical model theoretical semivariogram nugget effect in the $0^0$ direction.

*C0(90)=* - Specifies the spherical model theoretical semivariogram nugget effect in the $90^0$ direction.

*Spherical_Correlation(0)=* - Specifies the spherical model theoretical semivariogram simple correlation in the $0^0$ direction.

*Spherical_Correlation(90)=* - Specifies the spherical model theoretical semivariogram simple correlation in the $90^0$ direction.

*Linear_B0(0)=* - Specifies the linear model theoretical semivariogram $B0$ coefficient in the $0^0$ direction.

*Linear_B0(90)=* - Specifies the linear model theoretical semivariogram $B0$ coefficient in the $90^0$ direction.

*Linear_B1(0)=* - Specifies the linear model theoretical semivariogram $B1$ coefficient in the $0^0$ direction.

*Linear_B1(90)=* - Specifies the linear model theoretical semivariogram $B1$ coefficient in the $90^0$ direction.

*Linear_Correlation(0)=* - Specifies the linear model theoretical semivariogram simple correlation in the $0^0$ direction.

*Linear_Correlation(90)=* - Specifies the linear model theoretical semivariogram simple correlation in the $90^0$ direction.

*Dewijsian_B0(0)=* - Specifies the dewijsian model theoretical semivariogram $B0$ coefficient in the $0^0$ direction.

*Dewijsian_B0(90)=* - Specifies the dewijsian model theoretical semivariogram $B0$ coefficient in the $90^0$ direction.

*Dewijsian_B1(0)=* - Specifies the dewijsian model theoretical semivariogram $B1$ coefficient in the $0^0$ direction.

*Dewijsian_B1(90)=* - Specifies the dewijsian model theoretical semivariogram $B1$ coefficient in the $90^0$ direction.

*Dewijsian_Correlation(0)=* - Specifies the dewijsian model theoretical semivariogram simple correlation in the $0^0$ direction.

*Dewijsian_Correlation(90)=* - Specifies the dewijsian model theoretical semivariogram simple correlation in the $90^0$ direction.

*Xstep=* - Specifies the step size in the x direction for semivariogram calculation of gridded data. Only applies to the $Varfit$ program.

*Ystep*= - Specifies the step size in the y direction for semivariogram calculation of gridded data. Only applies to the *Varfit* program.

*Xmax*= - Specifies the maximum x coordinate value of the data set.

*Xmin*= - Specifies the minimum x coordinate value of the data set.

*Ymax*= - Specifies the maximum y coordinate value of the data set.

*Ymin*= - Specifies the minimum y coordinate value of the data set.

*Tolerance*= - Specifies the minimum distance between points to consider then as distinct. If two points are closer than this value only the first point is kept and the second point is discarded. This is done so that a singular matrix, in the kriging system of equations, does not result. Only applies to the *Krige* program.

*Maximum_Variance*= - Specifies the maximum error variance allowed in the selection of a minimal data set. Used only if minimizing. Based on this value the program calculates the following value based on the following equation:

$$Largest\_Difference = Confidence\_Interval \cdot \sqrt{Maximum\_Variance}$$

Only applies to the *Krige* program.

*Largest_Difference*= - Specifies the desired maximum difference between the estimated value and the actual value at each point. To be used in the selection of a minimal data set. Used only if minimizing. Based on this value the program calculates the following value based on the following equation:

$$Maximum\_Variance = \left( \frac{Largest\_Difference}{Confidence\_Interval} \right)^2$$

Only applies to the *Krige* program.

*Confidence_Interval*= - Specifies the desired statistical confidence interval. To be used in the selection of a minimal data set. Used only if minimizing. Based on this

value the program calculates the following values based on the following equations in the order shown:

$$Maximum\_Variance = Old\_Variance \cdot \left( \frac{Old\_Value}{Confidence\_Interval} \right)^2$$

$$Largest\_Difference = Confidence\_Interval \cdot \sqrt{Maximum\_Variance}$$

Where Old_Value is the prior value of Confidence_Interval and Old_Variance is the prior value of Maximum_Variance. Only applies to the *Krige* program.

*Polynomial=* - Specifies the five coefficients of the polynomial used for trend removal. The coefficients are speciffied for the following polynomial in the order shown:

$$A + Bx + Cy + Dxy + Ex^2 + Fy^2 = z$$

Only applies to the *Rebuild* program.

*Var_Psi=* - Specifies the semi-inclusion angle, regularization factor, for non-gridded data in the semivariogram calculations. If the angle of one point relative to a second point is in the desired direction for semivariogram calculation and the points are a multiple of Var_Step apart then that pair of points is used in the calculation. The angle is allowed to vary from the desired direction by Var_Psi. Only applies to the *Varfit* program.

*Var_Step=* - Specifies the step size for non-gridded data in the semivariogram calculations. If the angle of one point relative to a second point is in the desired direction for semivariogram calculation and the points are a multiple of Var_Step apart then that pair of points is used in the calculation. Only applies to the *Varfit* program.

*Number_Of_X_Increments=* - Specifies the number of grid points in the x direction at which to estimate surface values and estimation variances for the *Krige* program. For the other programs or if expanding the image using Expansion_Factor,

it is the number of grid points in the x direction of the original gridded data. If expanding the image only grid points between known points are added and not points beyond the edge of the data set. The kriging program has a default value of one for the expansion factor and calculates the number of expanded grid locations as follows:

$$Number\_Of\_X\_Increments = (Number\_Of\_X\_Increments - 1) \cdot Expansion\_Factor + 1$$

*Number_Of_Y_Increments*= - Specifies the number of grid points in the y direction at which to estimate surface values and estimation variances for the *Krige* program. For the other programs or if expanding the image using Expansion_Factor, it is the number of grid points in the y direction of the original gridded data. If expanding the image only grid points between known points are added and not points beyond the edge of the data set. The kriging program has a default value of one for the expansion factor and calculates the number of expanded grid locations as follows:

$$Number\_Of\_Y\_Increments = (Number\_Of\_Y\_Increments - 1) \cdot Expansion\_Factor + 1$$

*Minimizing* - Instructs the program to select a minimal data set. The initial data set is based on an "x" pattern using the semi-inclusion distances calculated from the range or set by the user. Only aplies to the *Krige* program.

*Integer_Data* - Informs program that data set is in integer format, otherwise data set is read in as floats.

*Total_Number_Of_Points*= - Specifies the total number of points in the data set. Not used in the *Krige* program.

*Gridded_Data* - Informs program that data set is gridded and to use that fact to drastically reduce the computational time required.

*Pattern_Output* - Instructs *Krige* program to print minimal data set pattern using an "x" for an included point and a "." for an excluded point.

*Angles=* - Specifies the angles, in degrees, at which to generate semivariograms. Only applies to the *Varfit* program.

*Maxlag=* - Specifies the maximum distance to use as termination criteria for semivariogram generation. Half this distance is then used in the determination of the theoretical semivarioogram models. Only applies to the *Varfit* program.

*IAX=* - Specifies the semi-inclusion distance for initial point inclusion for minimal data set generation or for inclusion in estimation for normal kriging in $x$ direction. If image is to be expanded then this applies to the original grid and the program multiplies this by the expansion factor. This allows the input to be based on the original image and changing the amount by which to expand the image is accomplished by changing only the expansion factor input. If not provided then this value is calculated as follows:

$$IAX = \frac{Number\_Of\_X\_Increments \cdot A(0)}{(Xmax - Xmin) \cdot \sqrt{2}}$$

Only applies to the *Krige* program.

*IAY=* - Specifies the semi-inclusion distance for initial point inclusion for minimal data set generation or for inclusion in estimation for normal kriging in $y$ direction. If image is to be expanded then this applies to the original grid and the program multiplies this by the expansion factor. This allows the input to be based on the original image and changing the amount by which to expand the image is accomplished by changing only the expansion factor input. If not provided then this

value is calculated as follows:

$$IAY = \frac{Number\_Of\_Y\_Increments \cdot A(0)}{(Ymax - Ymin) \cdot \sqrt{2}}$$

Only applies to the *Krige* program.

*Maximum_Points=* - Specifies the maximum number of points to allow an estimate to be based upon. Default is 250 due to unreliability of matrix inversion routine for values larger than 250. Only applies to *Krige* program.

*Var_Angle=* - Specifies the angle at which to generate a semivariogram. Only applies to the *Varfit* program.

*Expansion_Factor=* - Specifies the amount by which to expand the image in the $x$ and $y$ directions. Increases the number of x and y increments and the semi-inclusion distances as indicated under the sections of the same name. Only applies to the *Krige* program.

*Invert_Output* - Instructs program to output points by columns instead of by rows. Only applies to *Krige* program.

*Number_Of_Header_Lines=* - Informs program of how many lines to skip, as header lines, in the data file before reading data.

*Trend_Plus_Weights=* - Instructs program on what type of local drift to remove while kriging. Default value is three. If illegal value provided then default is used. Legal values are:

1 - Do not remove any trend,

3 - Remove linear trend, or

6 - Remove quadratic trend.

Only applies to *Krige* program.

*Use_Largest_Difference* - Instructs program to calculate actual, not theoretical, estimation variance based on the difference between the estimate and the actual value. Only applies to *Krige* program doing minimization.

*Do_Shebang* - Instructs the program to select minimal data set by minimizing the entire data set and not by using pattern replication. Much slower this way. Only applies to *Krige* program doing minimization.

*Print_Interval=* - Instructs the program on how often the "*Status.Report*" file is to be updated. Can save considerable time since program may be input/output bound doing normal kriging or minimizing not using pattern replication. The value of the input determines how many points are estimated before the status file is updated. Only applies to *Krige* program.

*A.2 Example Krige Input Files*

```
Values different for each run

Maximum_Variance= 85.0
Minimal_Dataset_Filename=kriged85.min
Output_Filename=kriged85.surf
Variance_Filename=kriged85.variance
Error_Filename=kriged85.out

Common values for all runs on this dataset

A(0)= 102.994, C(0)= 571.15, C0(0)= 32.231
A(90)= 95.117, C(90)= 312.925, C0(90)= 75.038
Print_Interval= 25

Common values for all runs on all datasets

Xmin= 0.0, Xmax= 266.0, Ymin= 0.0, Ymax= 267.0
Number_Of_X_Increments= 267, Number_Of_Y_Increments= 268
Minimizing
Pattern_Output
Gridded_Data
Integer_Data
Data_Filename=mri.data
End_Of_Input_Values
```

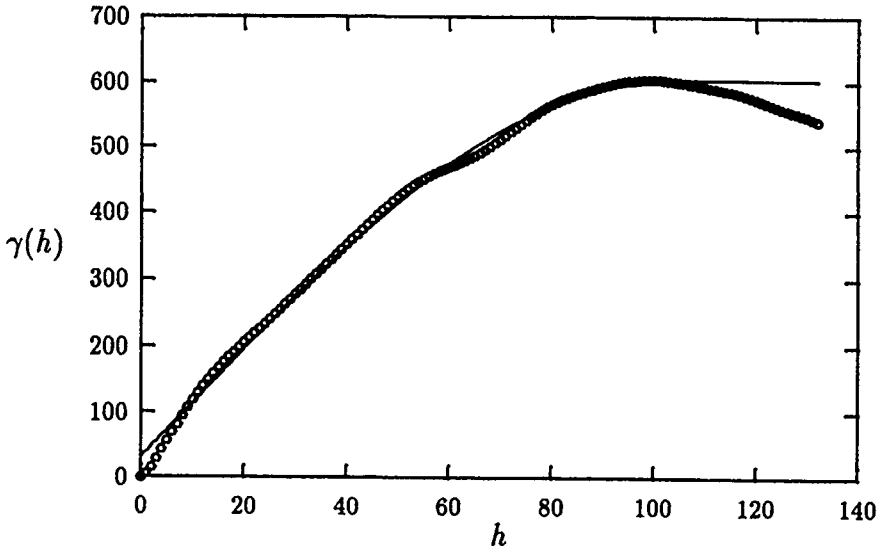
```
Values different for each run

Largest_Difference= 300.0
Minimal_Dataset_Filename=kriged3.min
Output_Filename=kriged3.surf
Variance_Filename=kriged3.variance
Error_Filename=kriged3.out
IAX= 3, IAY= 3

Common values for all runs on this dataset

A(0)= 117.257, C(0)= 814.124, C0(0)= 0.0
A(90)= 96.703, C(90)= 328.165, C0(90)= 62.293

Common values for all runs on all datasets

Xmin= 0.0, Xmax= 266.0, Ymin= 0.0, Ymax= 267.0
Number_Of_X_Increments= 267, Number_Of_Y_Increments= 268
Minimizing
Gridded_Data
Integer_Data
Data_Filename=mri.data
End_Of_Input_Values
```

*A.3 Example Varfit Input Files*

```
Xmin= 0.0, Xmax= 266.0, Ymin= 0.0, Ymax= 267.0
Number_Of_X_Increments= 267, Number_Of_Y_Increments= 268
Total_Number_Of_Points= 71556
Gridded_Data
Integer_Data
Data_Filename=mri.data
Output_Filename=../mri.repeat
Variogram_Filename=mri.vario
Plot_Filename=mri.plot
Xstep= 1.0, Ystep= 1.0, Var_Step= 1.0
MaxLag= 380
Error_Filename=mri.out
End_Of_Input_Values
```

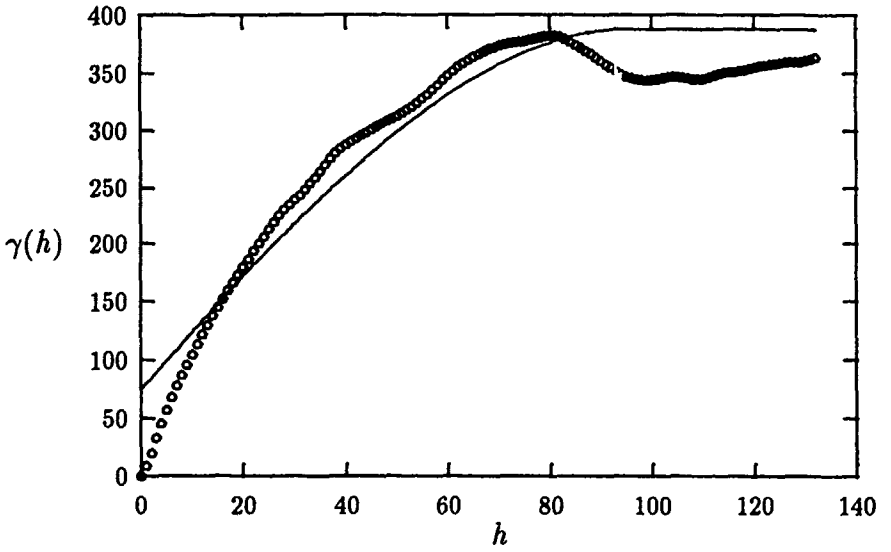
# Appendix B. *Semivariogram Plots*

This appendix includes the experimental and spherical theoretical Semivariogram plots for the $0^0$ and $90^0$ Directions for each slice of MRI data used in the kriging procedures.



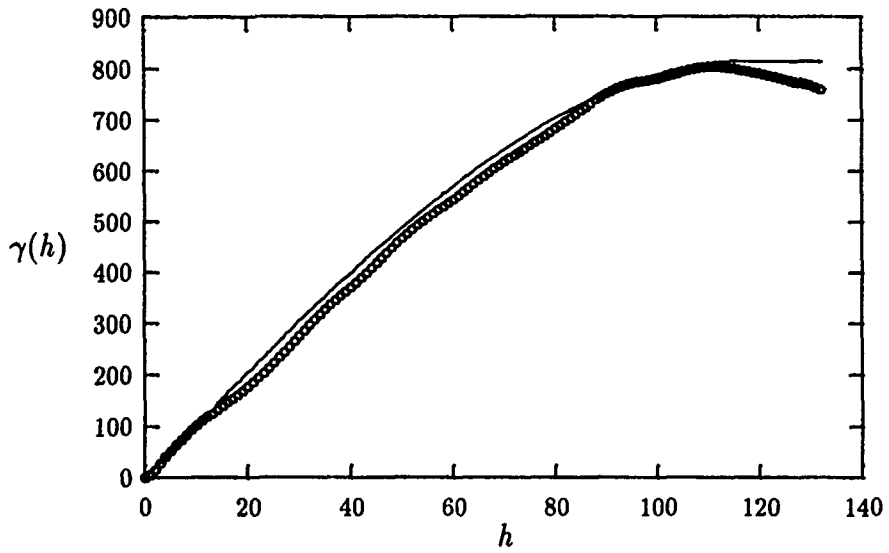

$$\gamma(h) = 571.150\left[\left(\frac{3}{2}\right)\left(\frac{h}{102.994}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{102.994}\right)^3\right] + 32.231$$

Figure B.1. Slice 1 Semivariogram in the $0^0$ Direction



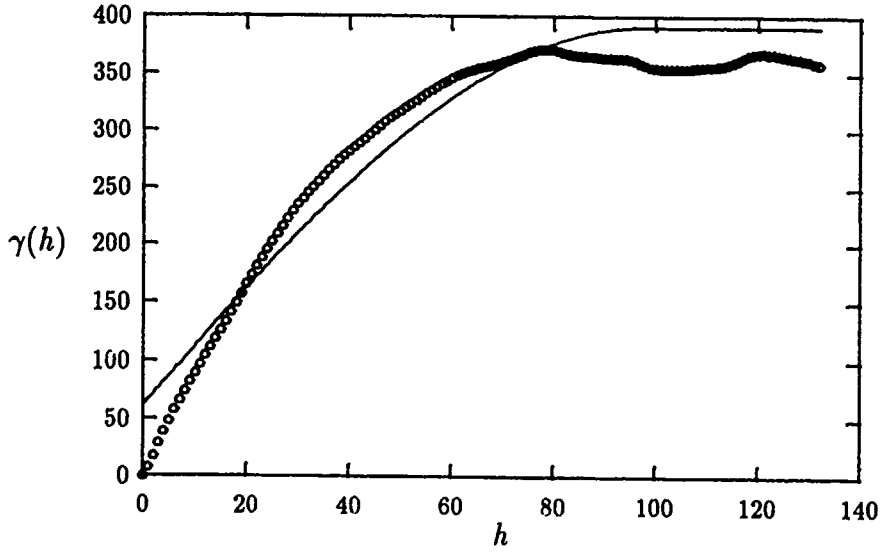

$$\gamma(h) = 312.925\left[\left(\frac{3}{2}\right)\left(\frac{h}{95.117}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{95.117}\right)^3\right] + 75.038$$

Figure B.2. Slice 1 Semivariogram in the $90^0$ Direction



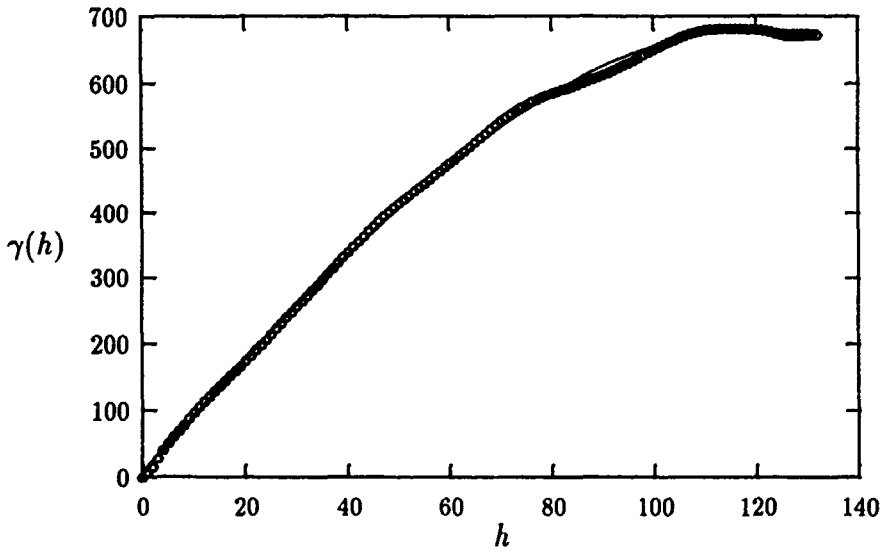

$$\gamma(h) = 814.124\left[\left(\frac{3}{2}\right)\left(\frac{h}{117.257}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{117.257}\right)^3\right] + 0.000$$

Figure B.3. Slice 3 Semivariogram in the $0^0$ Direction



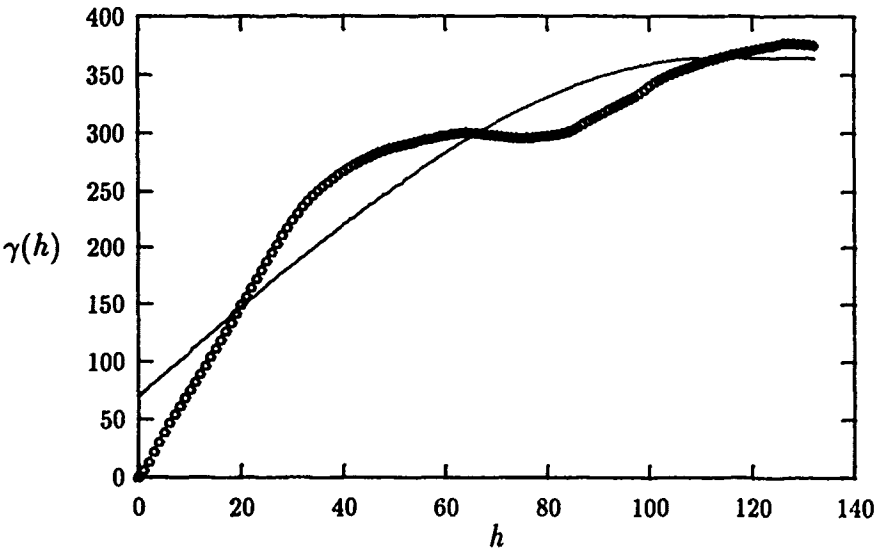

$$\gamma(h) = 328.165\left[\left(\frac{3}{2}\right)\left(\frac{h}{96.703}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{96.703}\right)^3\right] + 62.293$$

Figure B.4. Slice 3 Semivariogram in the $90^0$ Direction



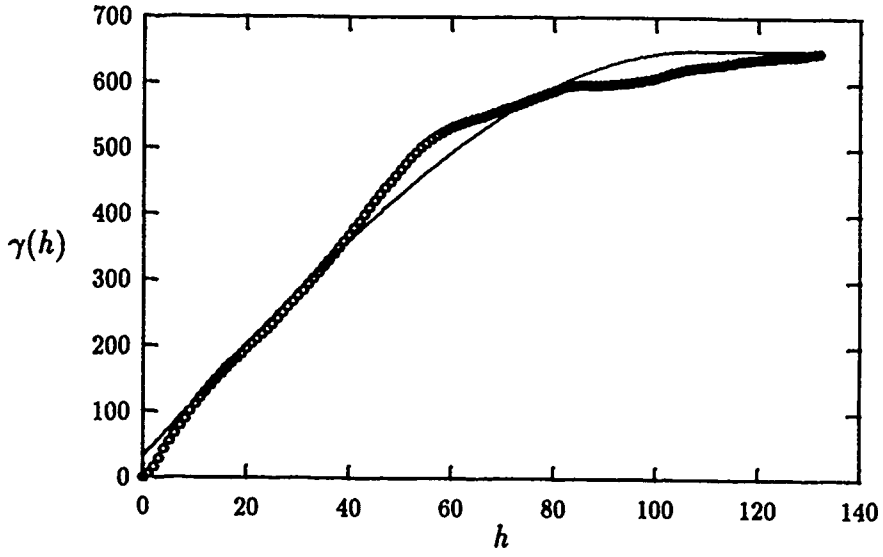

$$\gamma(h) = 669.468\left[\left(\frac{3}{2}\right)\left(\frac{h}{118.231}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{118.231}\right)^3\right] + 9.390$$

Figure B.5. Slice 6 Semivariogram in the $0^0$ Direction



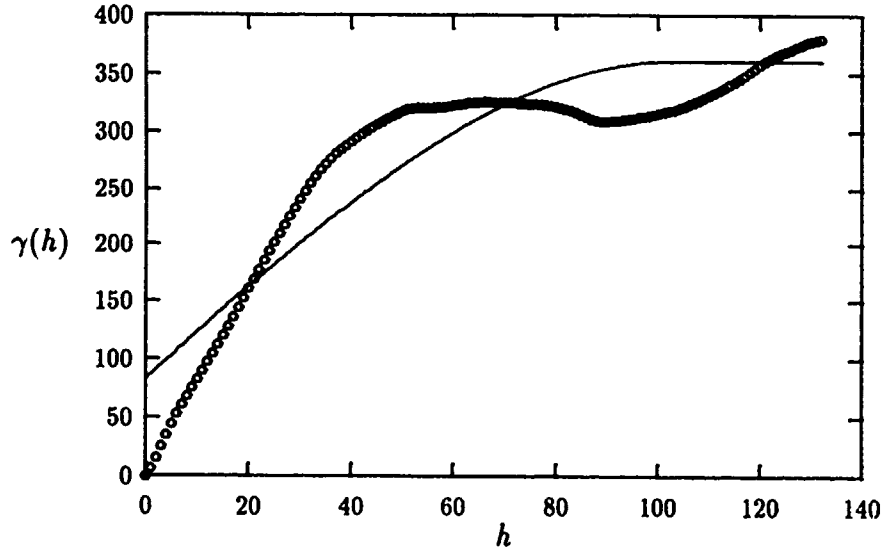

$$\gamma(h) = 294.044\left[\left(\frac{3}{2}\right)\left(\frac{h}{112.439}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{112.439}\right)^3\right] + 70.261$$

Figure B.6. Slice 6 Semivariogram in the $90^0$ Direction



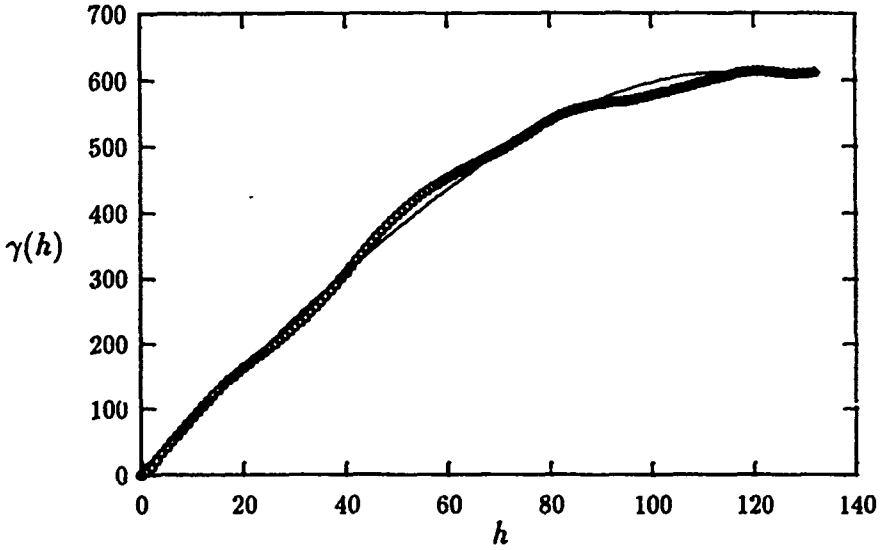

$$\gamma(h) = 616.325\left[\left(\frac{3}{2}\right)\left(\frac{h}{107.617}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{107.617}\right)^3\right] + 32.990$$

Figure B.7. Slice 9 Semivariogram in the $0^0$ Direction



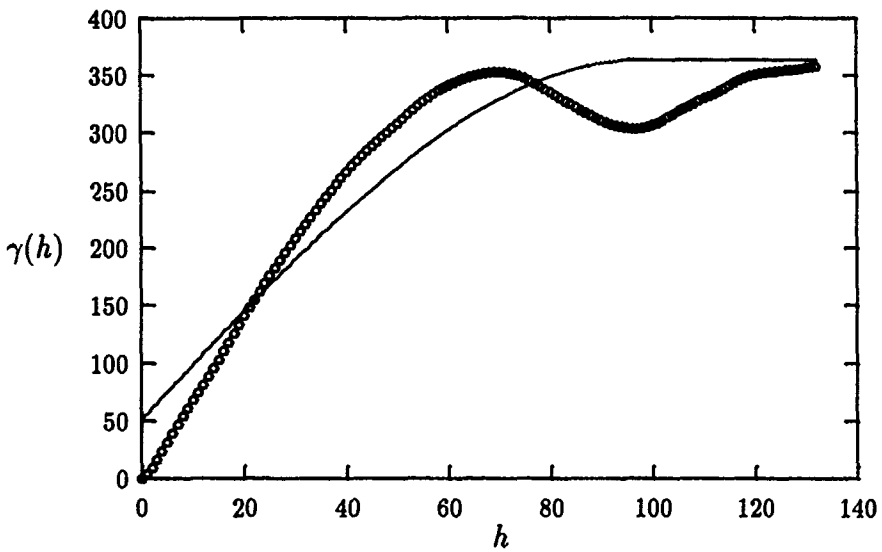

$$\gamma(h) = 277.228\left[\left(\frac{3}{2}\right)\left(\frac{h}{102.237}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{102.237}\right)^3\right] + 83.280$$

Figure B.8. Slice 9 Semivariogram in the $90^0$ Direction



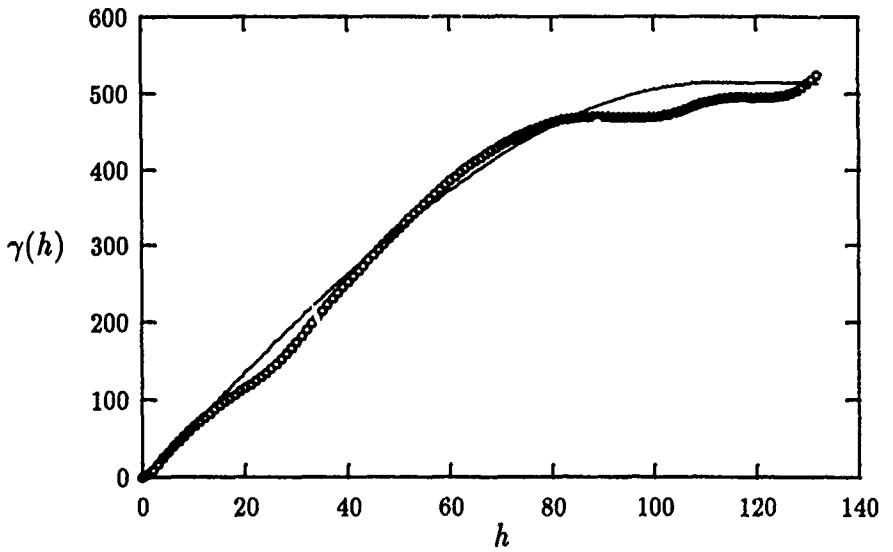

$$\gamma(h) = 602.753\left[\left(\frac{3}{2}\right)\left(\frac{h}{114.949}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{114.949}\right)^3\right] + 8.171$$

Figure B.9. Slice 12 Semivariogram in the $0^0$ Direction



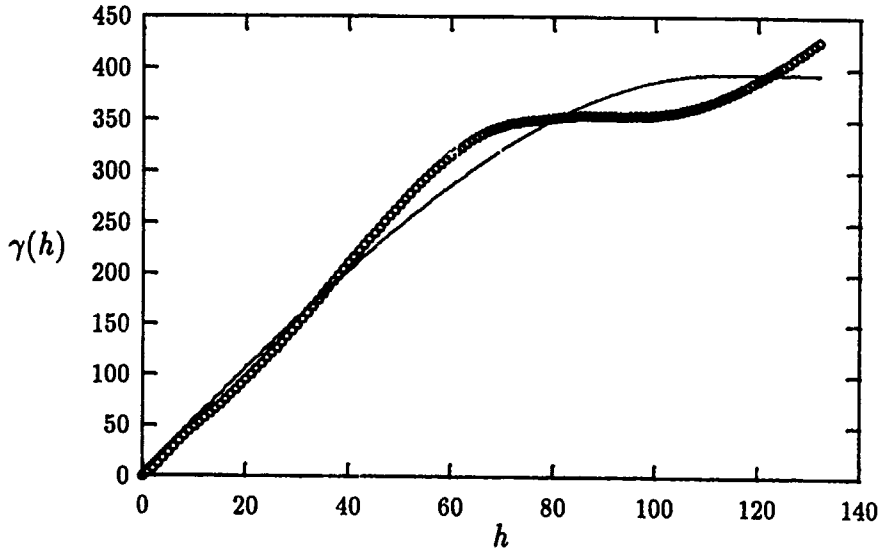

$$\gamma(h) = 311.715\left[\left(\frac{3}{2}\right)\left(\frac{h}{97.840}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{97.840}\right)^3\right] + 51.756$$

Figure B.10. Slice 12 Semivariogram in the $90^0$ Direction



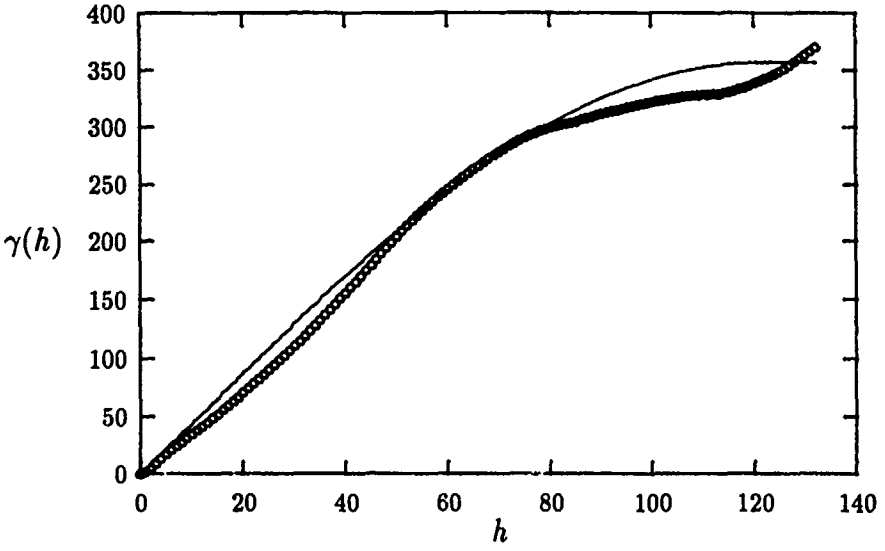

$$\gamma(h) = 514.049\left[\left(\frac{3}{2}\right)\left(\frac{h}{111.866}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{111.866}\right)^3\right] + 0.000$$

Figure B.11. Slice 15 Semivariogram in the $0^0$ Direction



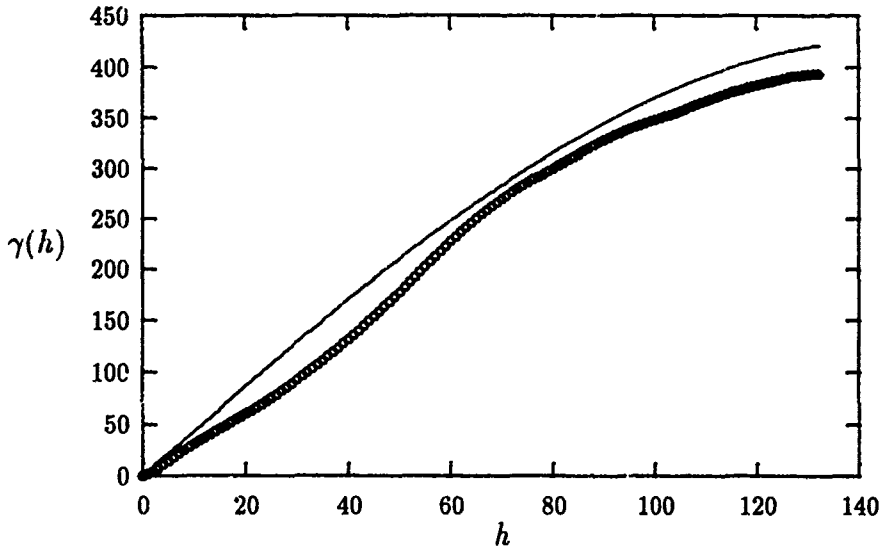

$$\gamma(h) = 389.894 \left[ \left(\frac{3}{2}\right) \left(\frac{h}{112.476}\right) - \left(\frac{1}{2}\right) \left(\frac{h}{112.476}\right)^3 \right] + 4.586$$

Figure B.12. Slice 15 Semivariogram in the $90^0$ Direction



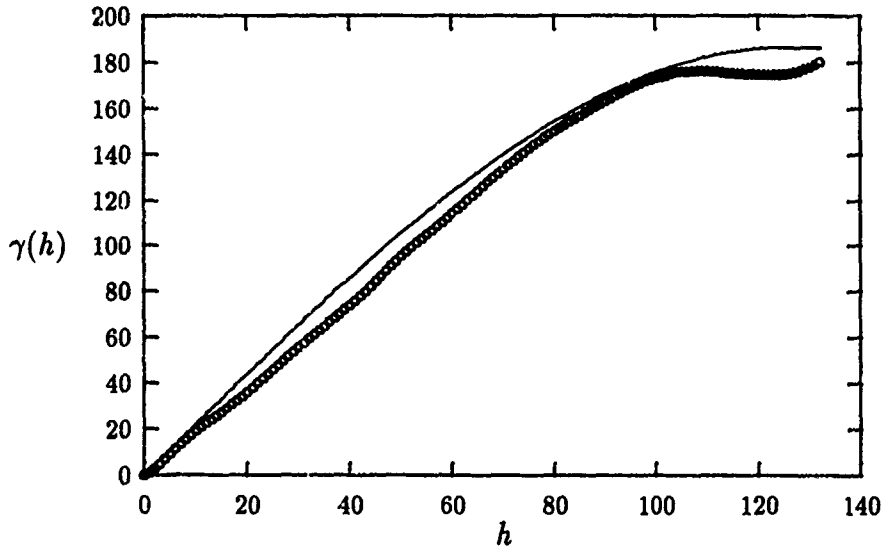

$$\gamma(h) = 356.499\left[\left(\frac{3}{2}\right)\left(\frac{h}{120.597}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{120.597}\right)^3\right] + 0.000$$

Figure B.13. Slice 18 Semivariogram in the $0^0$ Direction



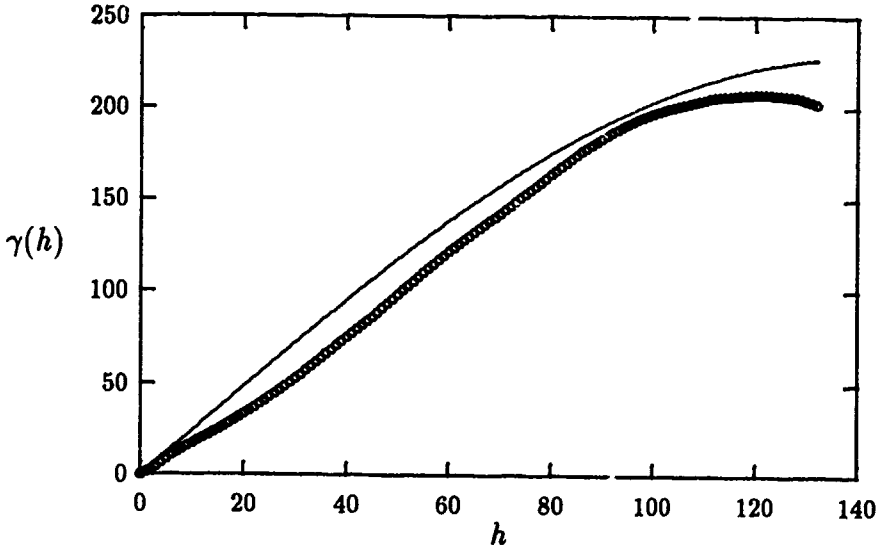

$$\gamma(h) = 427.220\left[\left(\frac{3}{2}\right)\left(\frac{h}{146.043}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{146.043}\right)^3\right] + 0.000$$

Figure B.14. Slice 18 Semivariogram in the $90^0$ Direction



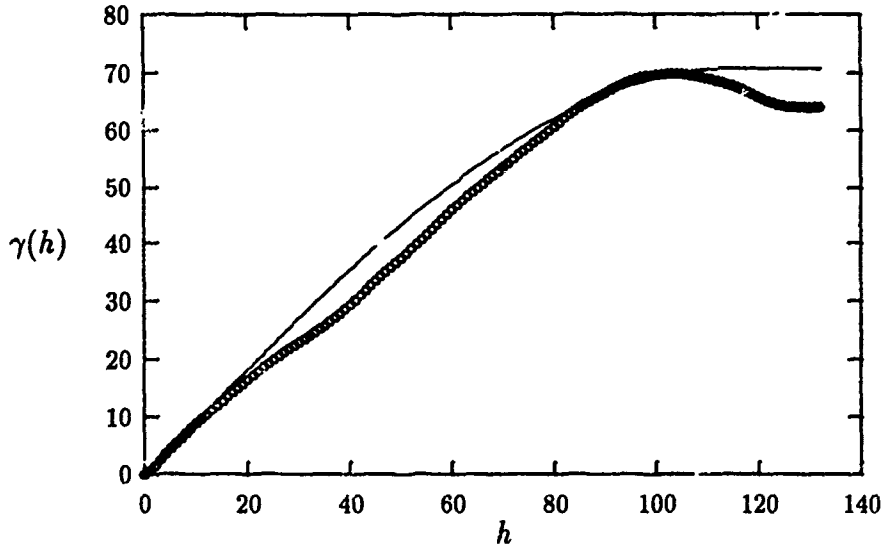

$$\gamma(h) = 186.436\left[\left(\frac{3}{2}\right)\left(\frac{h}{125.035}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{125.035}\right)^3\right] + 0.000$$

Figure B.15. Slice 21 Semivariogram in the $0^0$ Direction



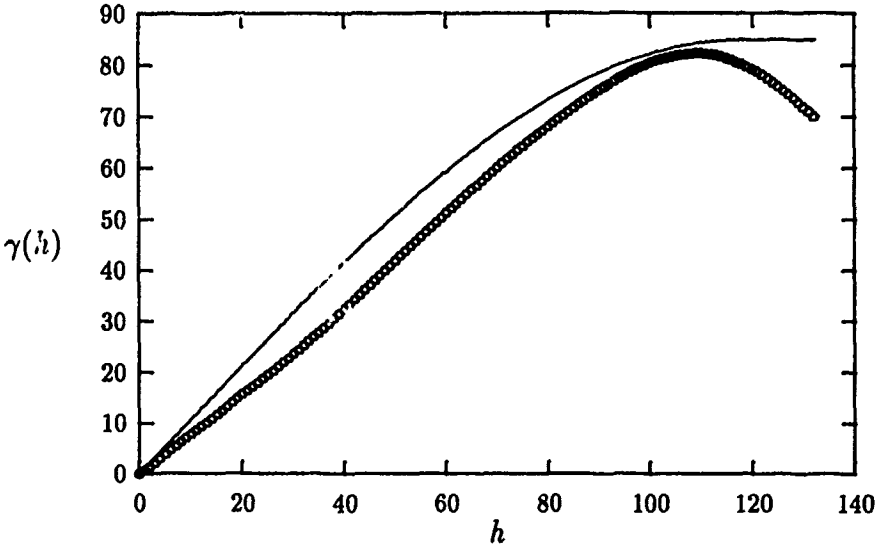

$$\gamma(h) = 227.803\left[\left(\frac{3}{2}\right)\left(\frac{h}{139.316}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{139.316}\right)^3\right] + 0.000$$

Figure B.16. Slice 21 Semivariogram in the $90^0$ Direction



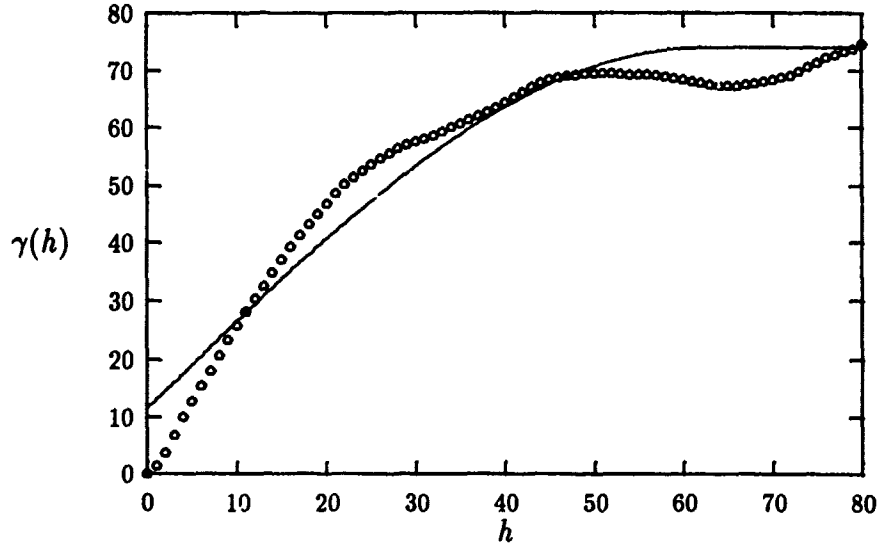

$$\gamma(h) = 70.764\left[\left(\frac{3}{2}\right)\left(\frac{h}{114.513}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{114.513}\right)^3\right] + 0.000$$

Figure B.17. Slice 24 Semivariogram in the $0^0$ Direction



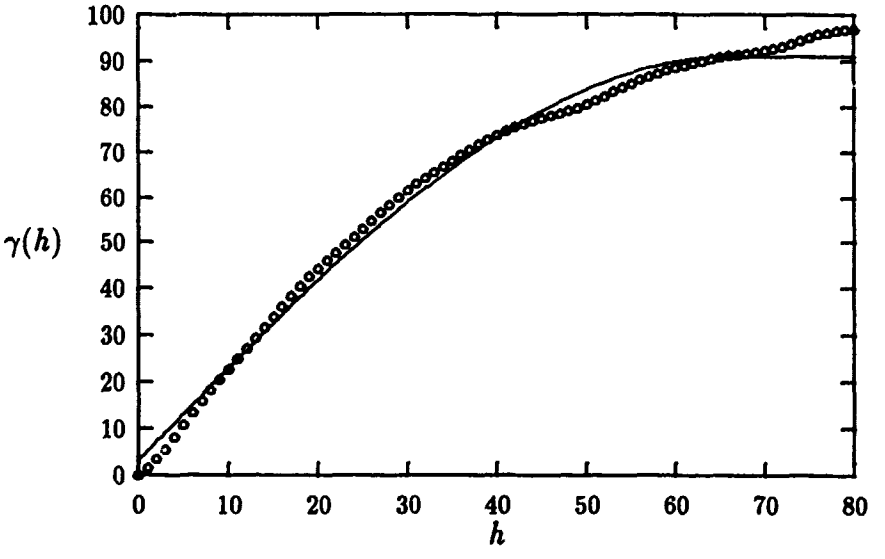

$$\gamma(h) = 24.869\left[\left(\frac{3}{2}\right)\left(\frac{h}{117.408}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{117.408}\right)^3\right] + 0.000$$

Figure B.18. Slice 24 Semivariogram in the $90^0$ Direction



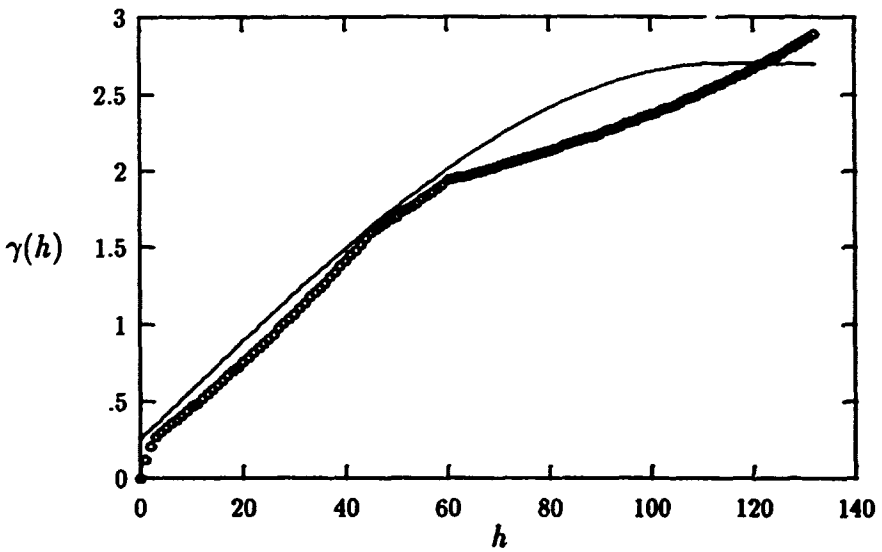

$$\gamma(h) = 62.243\left[\left(\frac{3}{2}\right)\left(\frac{h}{61.888}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{61.888}\right)^3\right] + 11.699$$

Figure B.19. Slice 24 Center Partition Semivariogram in the $0^0$ Direction



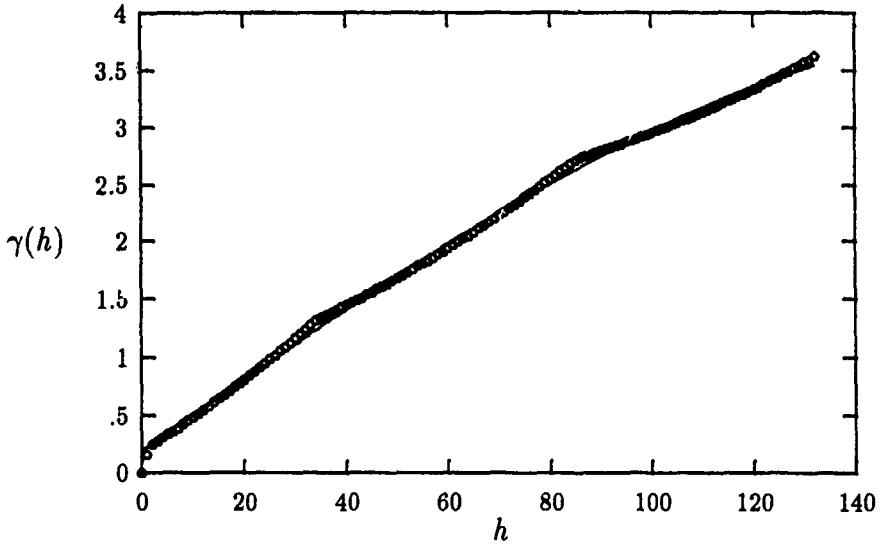

$$\gamma(h) = 87.601\left[\left(\frac{3}{2}\right)\left(\frac{h}{65.940}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{65.940}\right)^3\right] + 3.344$$

Figure B.20. Slice 24 Center Partition Semivariogram in the $90^0$ Direction




$$\gamma(h) = 2.443\left[\frac{3}{2}\left(\frac{h}{113.477}\right) - \frac{1}{2}\left(\frac{h}{113.477}\right)^3\right] + 0.258$$

Figure B.21. Slice 24 Background Partition Semivariogram in the $0^0$ Direction




$$\gamma(h) = 3.705\left[\left(\frac{3}{2}\right)\left(\frac{h}{180.109}\right) - \left(\frac{1}{2}\right)\left(\frac{h}{180.109}\right)^3\right] + 0.202$$

Figure B.22. Slice 24 Background Partition Semivariogram in the $90^0$ Direction

# Appendix C. *Photographs*

This appendix contains the photograghs showing the results of the minimization process. Each photogragh contains a series of images. Each series of images has the original data set on the left, the kriged image in the center, and the error between the original and kriged image on the right.

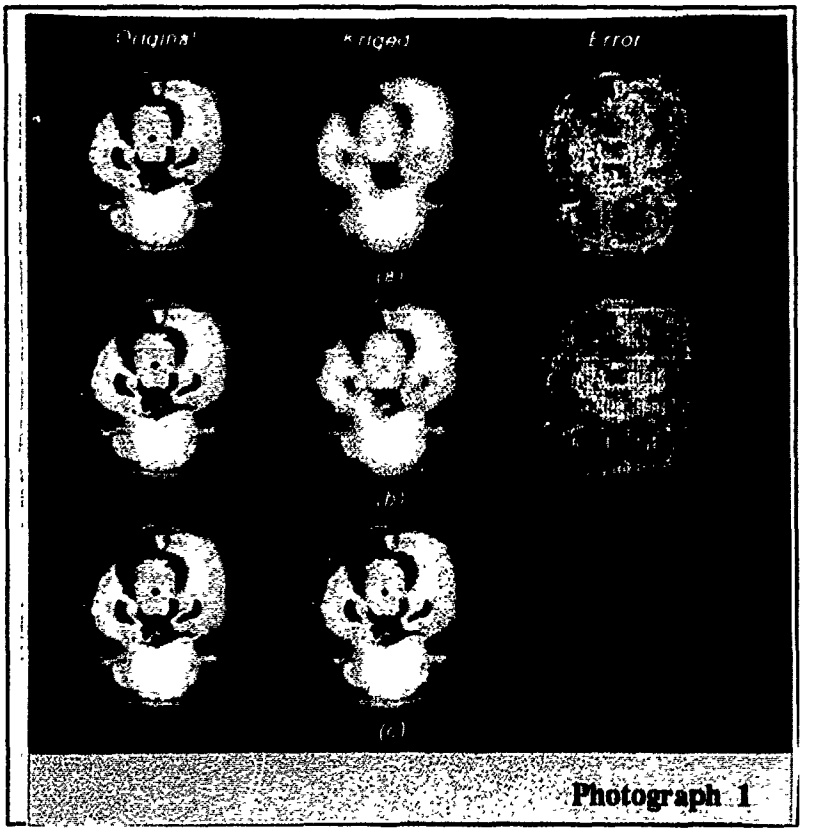



Figure C.1. Slice 1, Maximum Variance = 100, 85, 70

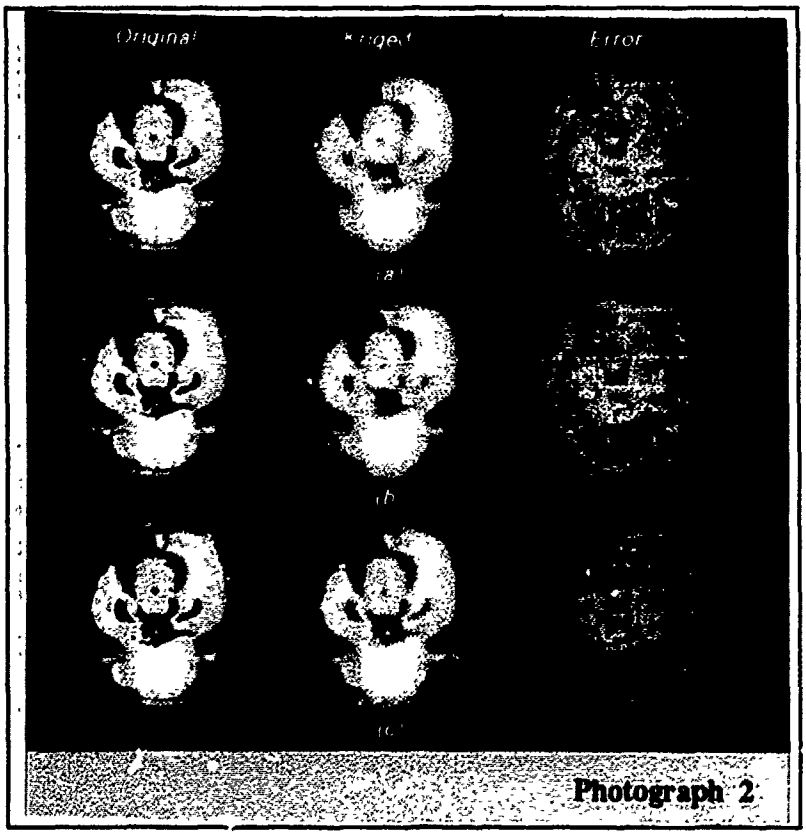



Figure C.2. Slice 1, Semi-Inclusion Distance = 5, 4, 3

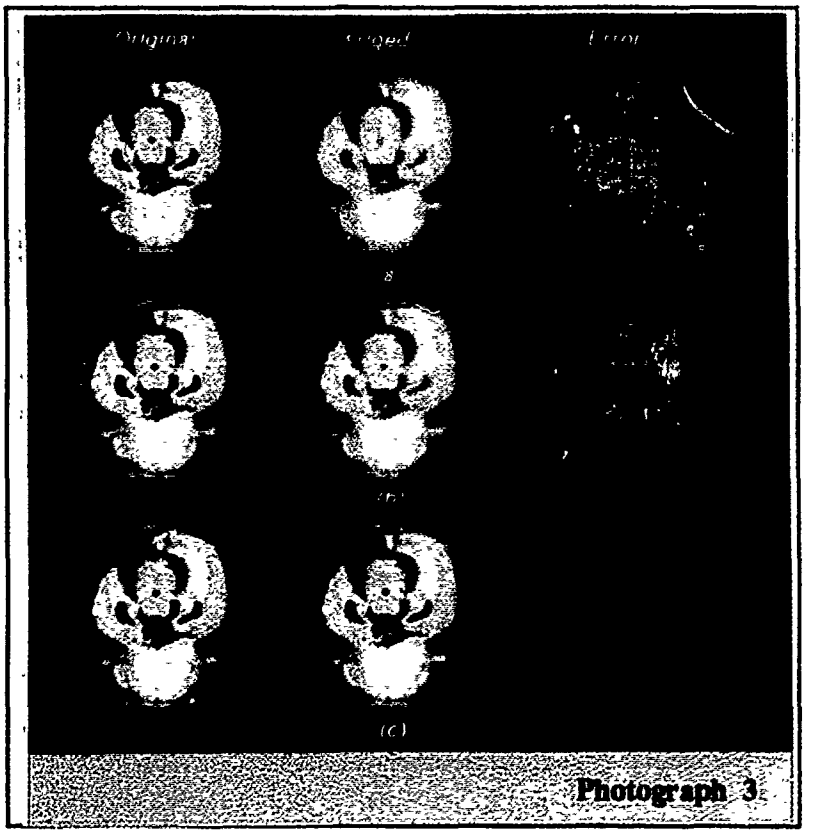



Figure C.3. Slice 1, Semi-Inclusion Distance = 3, 2, 1

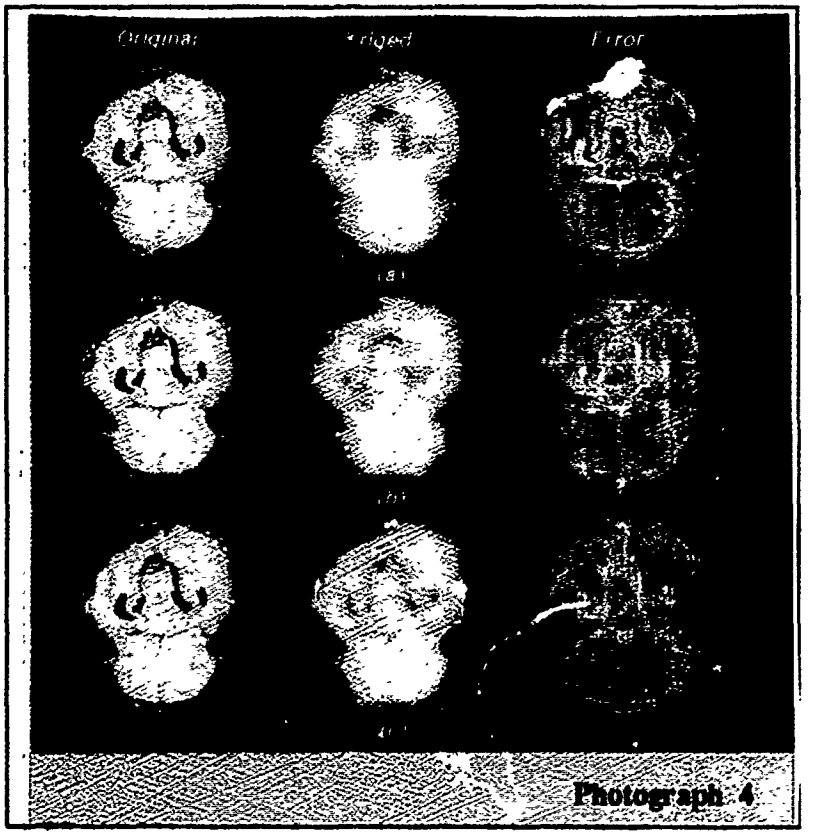



Figure C.4. Slice 3, Maximum Variance = 100, 85, 70

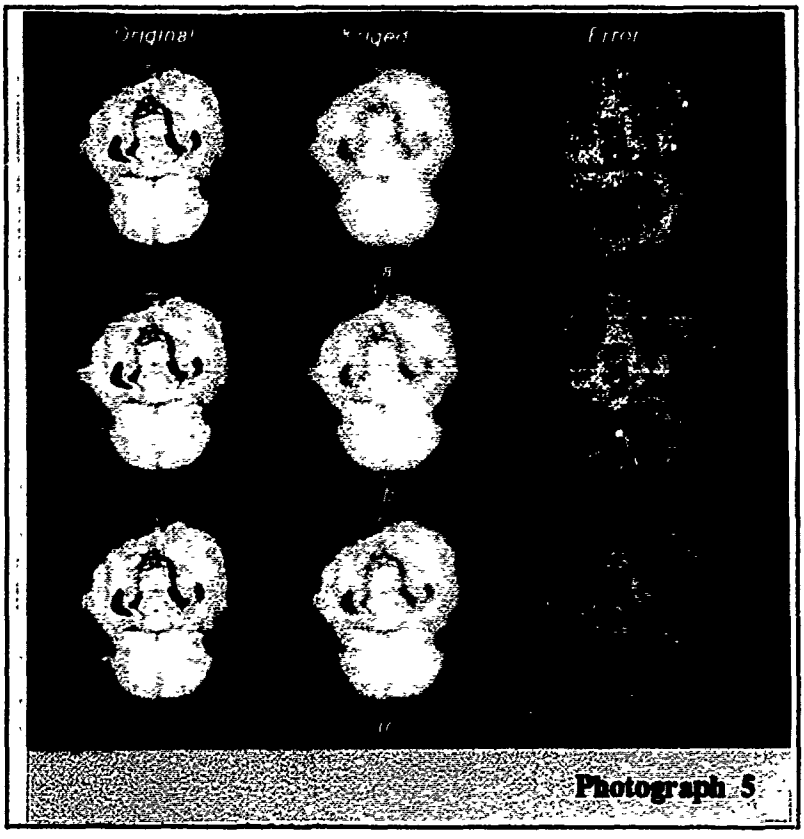



Figure C.5. Slice 3, Semi-Inclusion Distance = 5, 4, 3

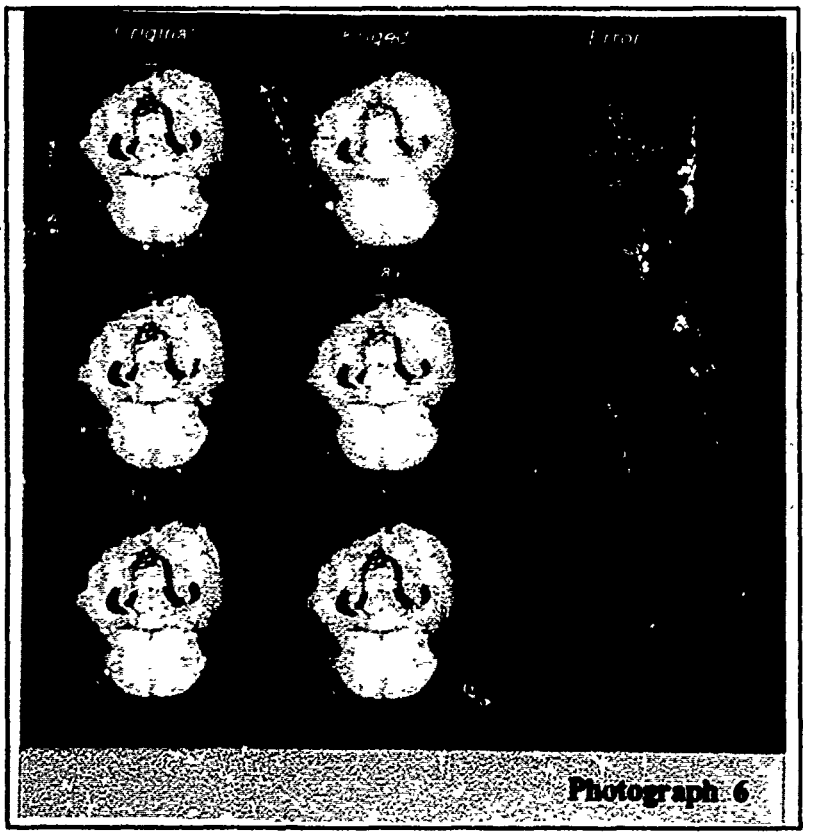



Figure C.6. Slice 3, Semi-Inclusion Distance = 3, 2, 1

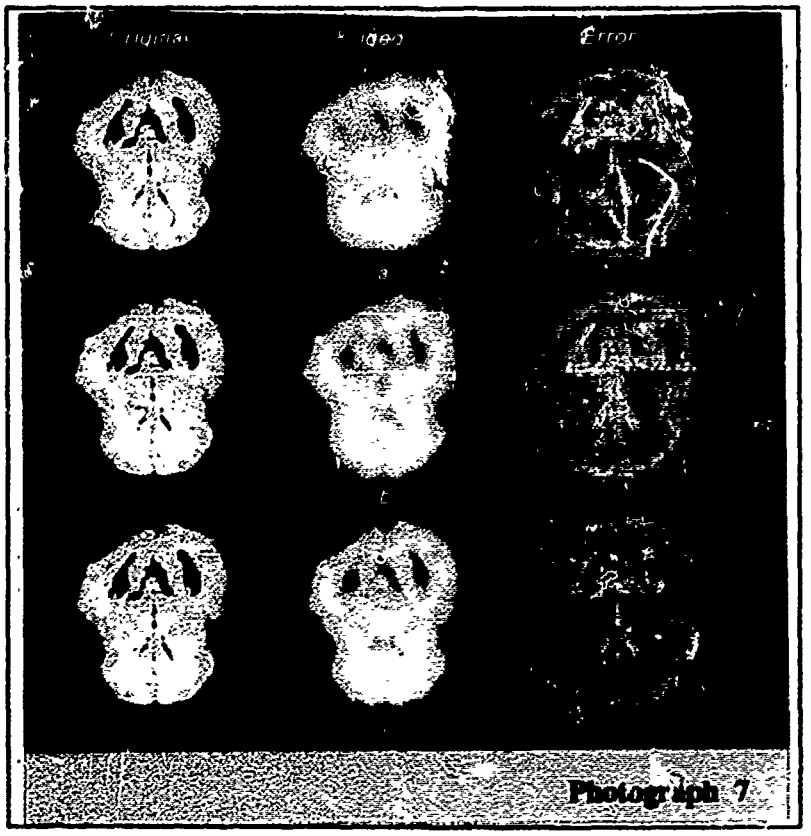



Figure C.7. Slice 6, Maximum Variance = 100, 85, 70

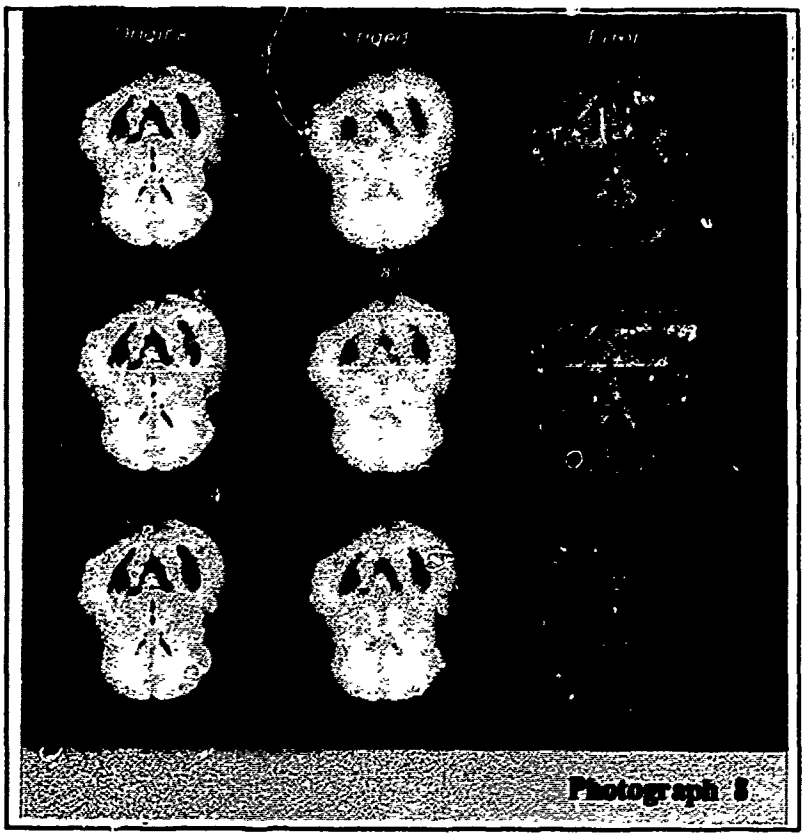



Figure C.8. Slice 6, Semi-Inclusion Distance = 5, 4, 3

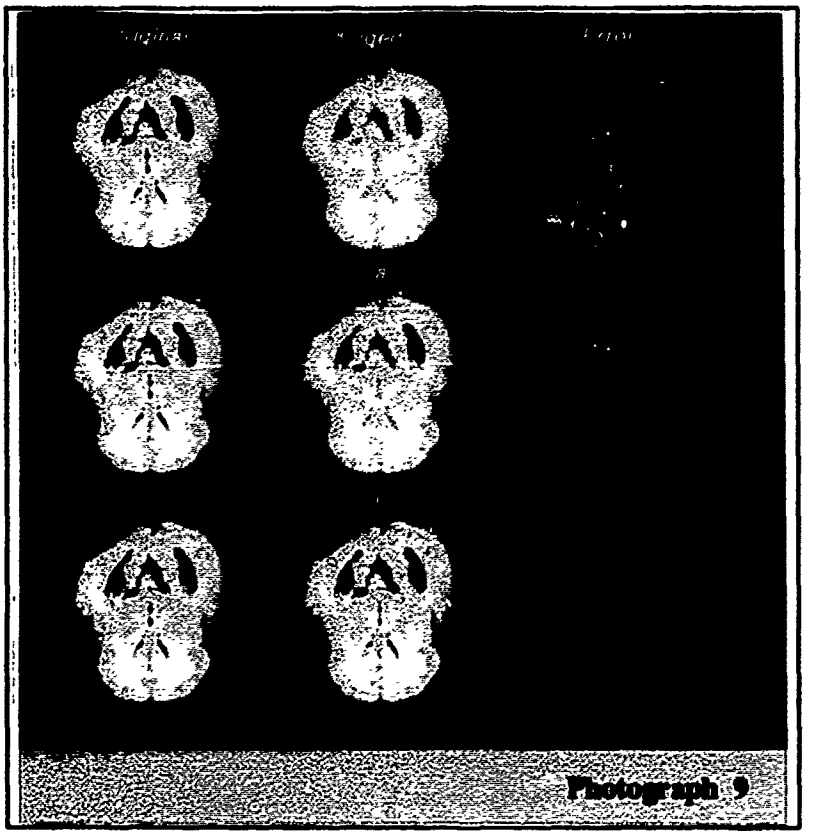



Figure C.9. Slice 6, Semi-Inclusion Distance = 3, 2, 1

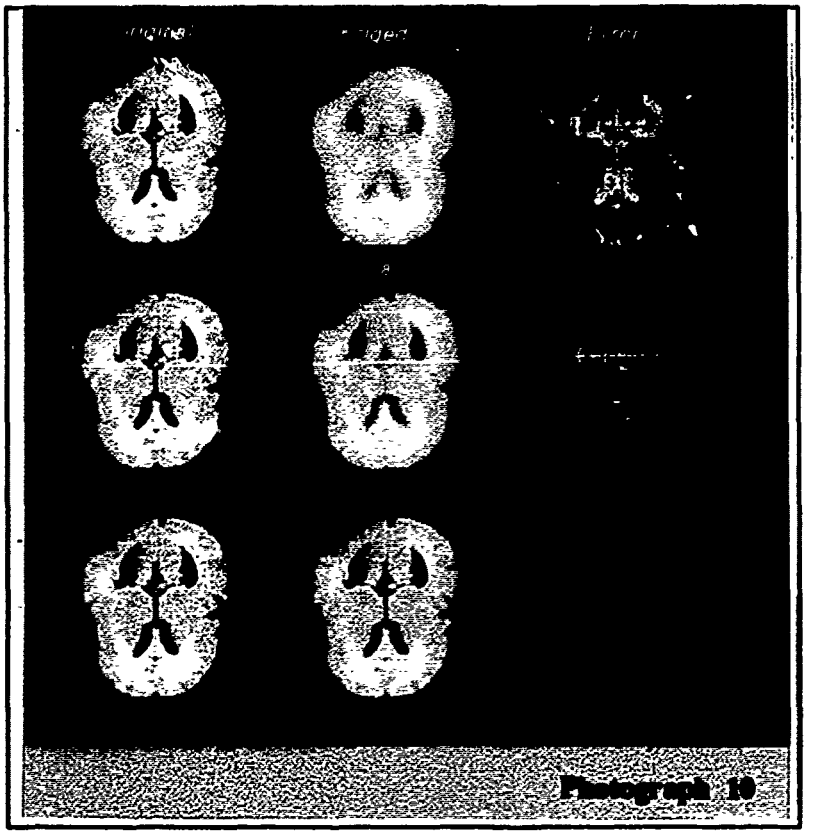


Figure C.10. Slice 9, Maximum Variance = 100, 85, 70

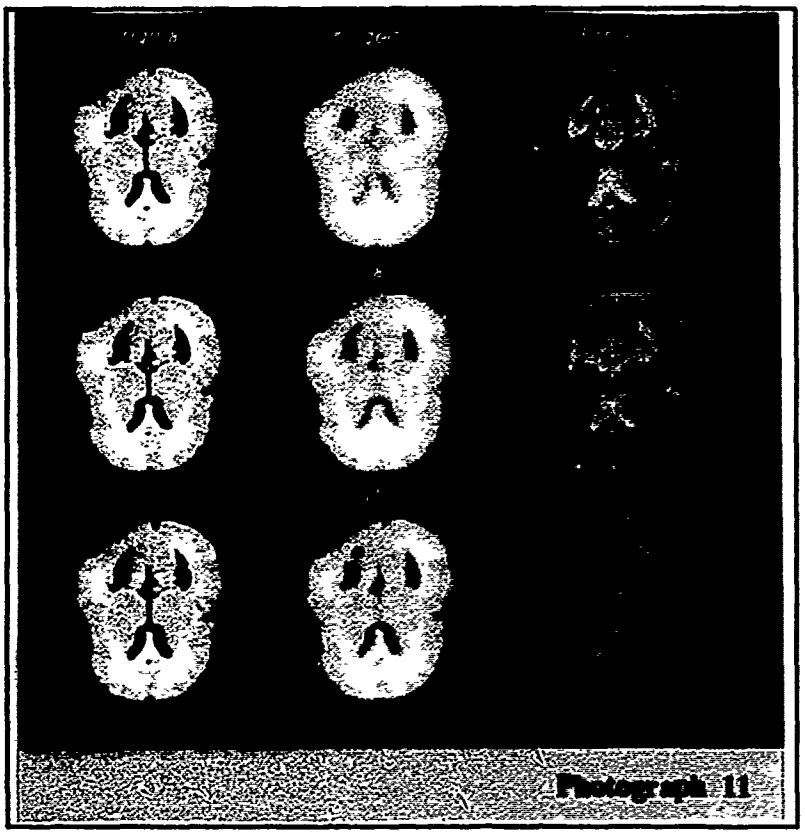


Figure C.11. Slice 9, Semi-Inclusion Distance = 5, 4, 3

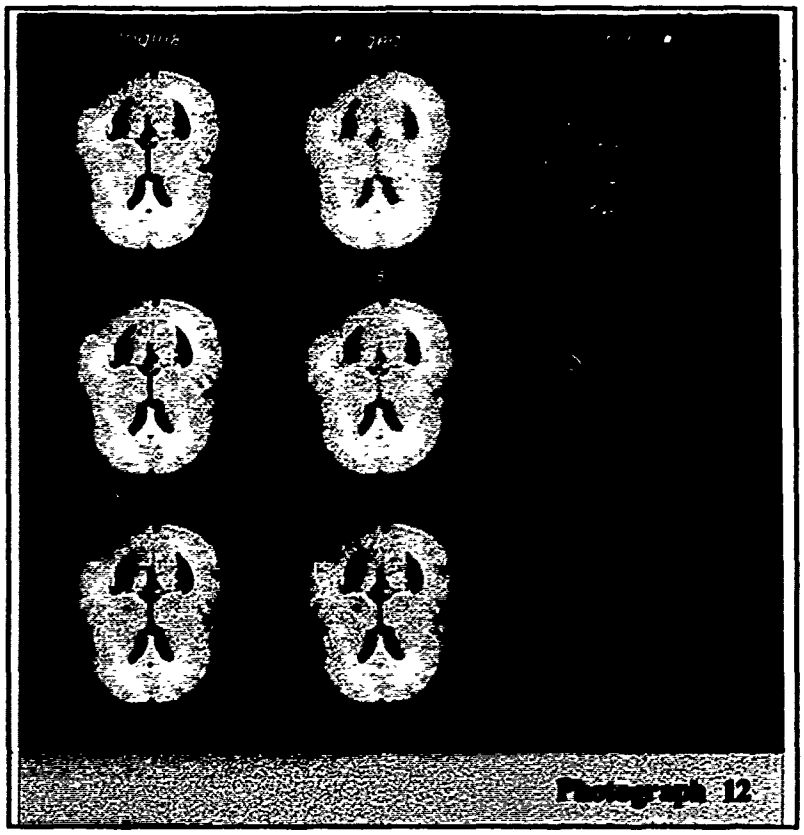


Figure C.12. Slice 9, Semi-Inclusion Distance = 3, 2, 1

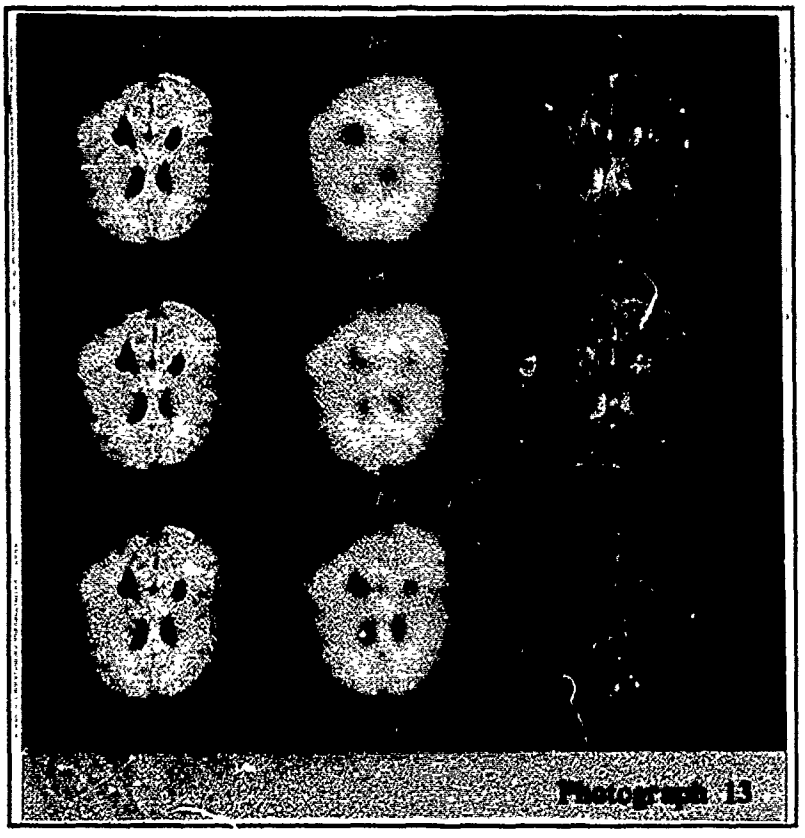



Figure C.13. Slice 12, Maximum Variance = 100, 85, 70

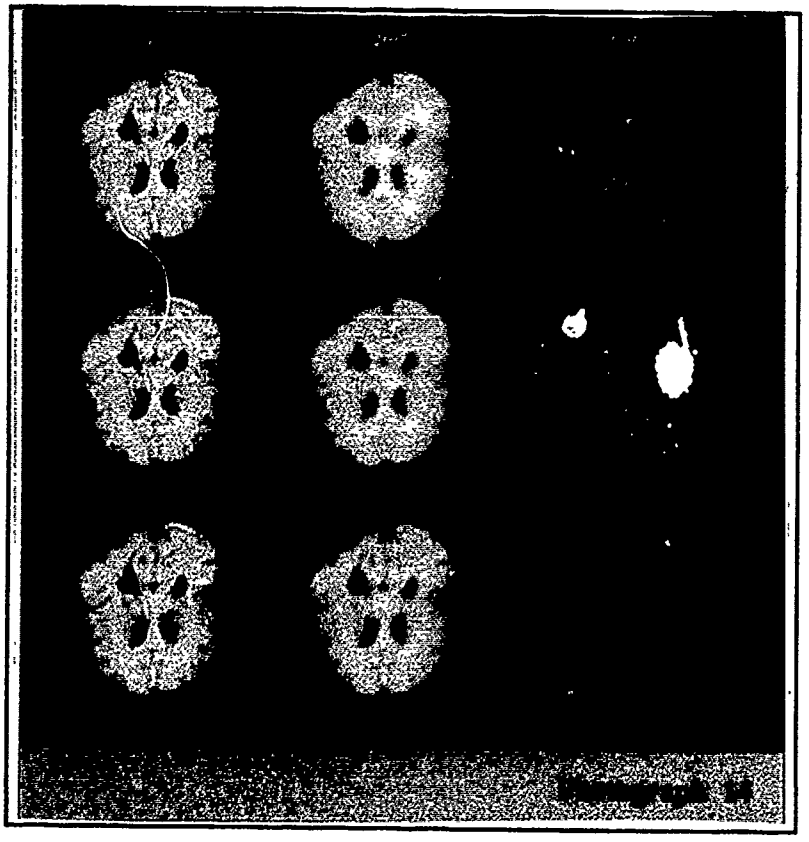


Figure C.14. Slice 12, Semi-Inclusion Distance = 5, 4, 3

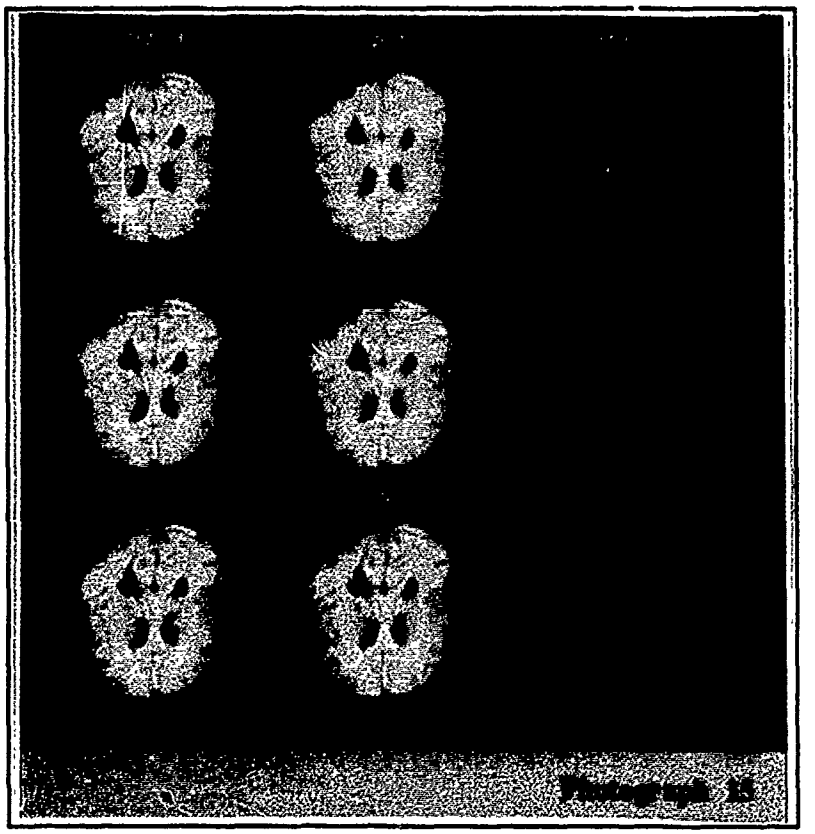


Figure C.15. Slice 12, Semi-Inclusion Distance = 3, 2, 1

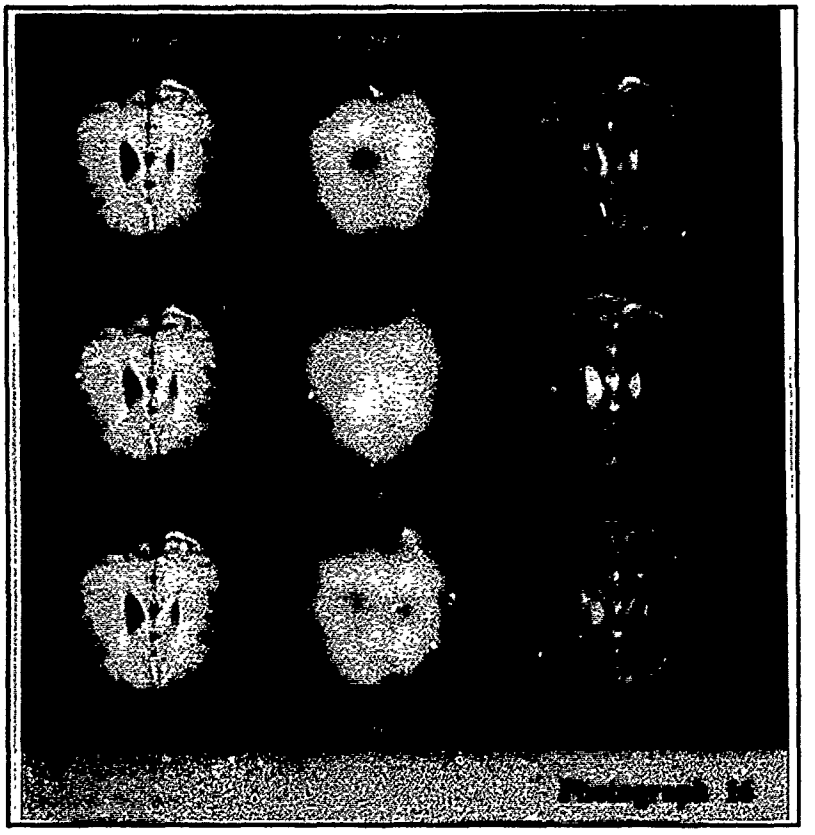


Figure C.16. Slice 15, Maximum Variance = 100, 85, 70

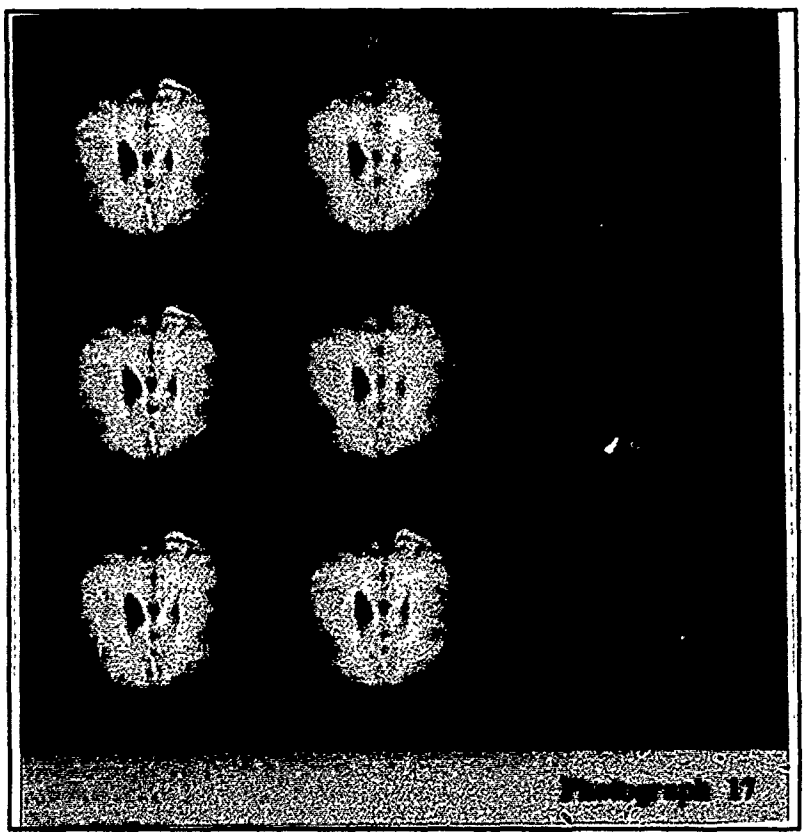


Figure C.17. Slice 15, Semi-Inclusion Distance = 5, 4, 3

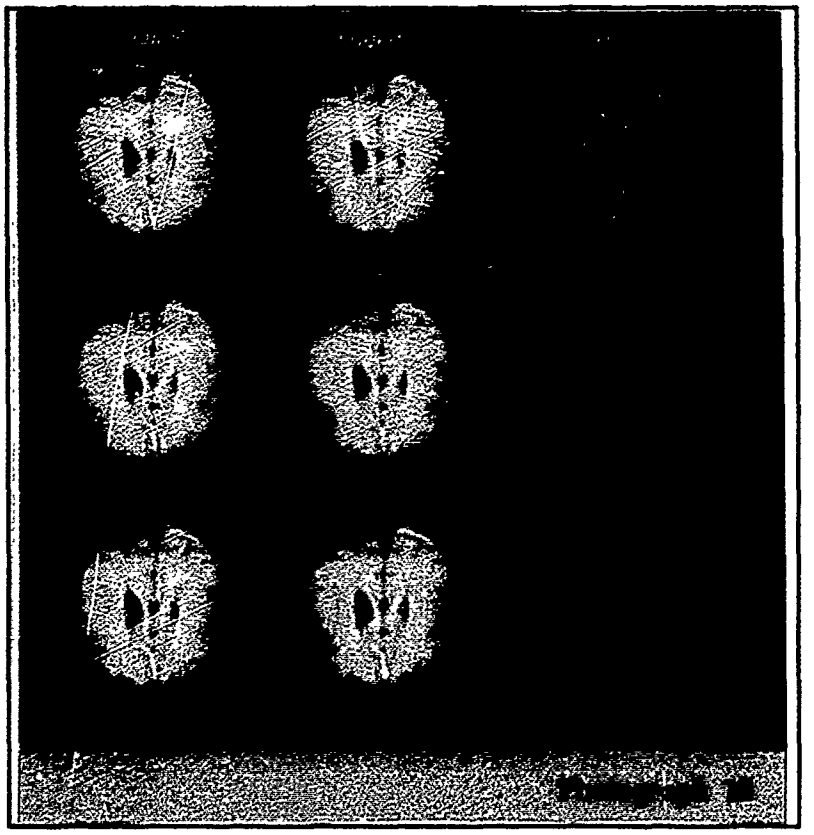


Figure C.18. Slice 15, Semi-Inclusion Distance = 3, 2, 1

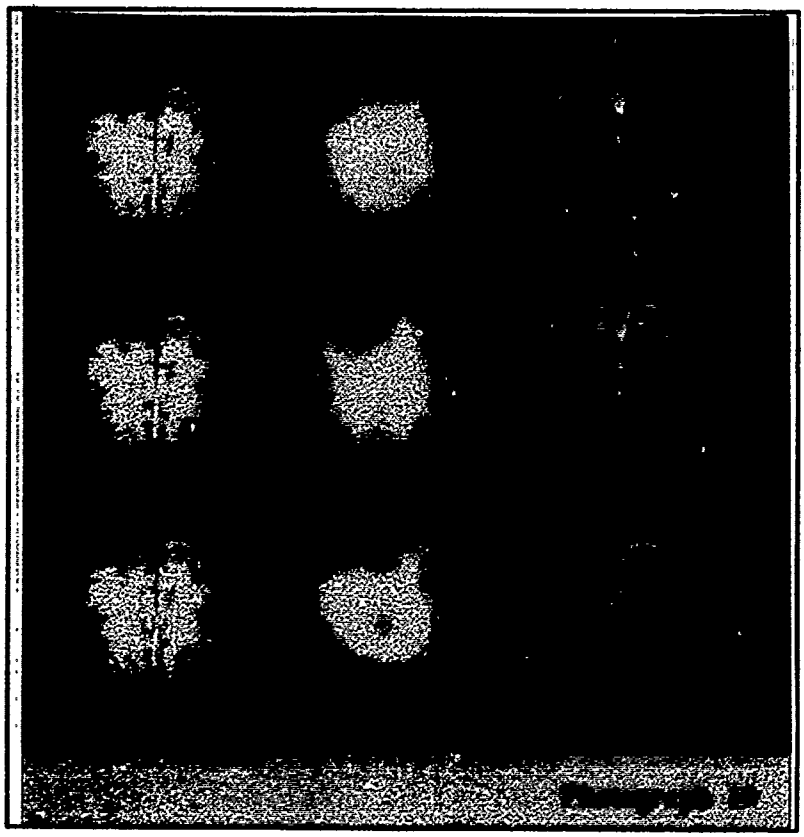


Figure C.19. Slice 18, Maximum Variance = 100, 85, 70

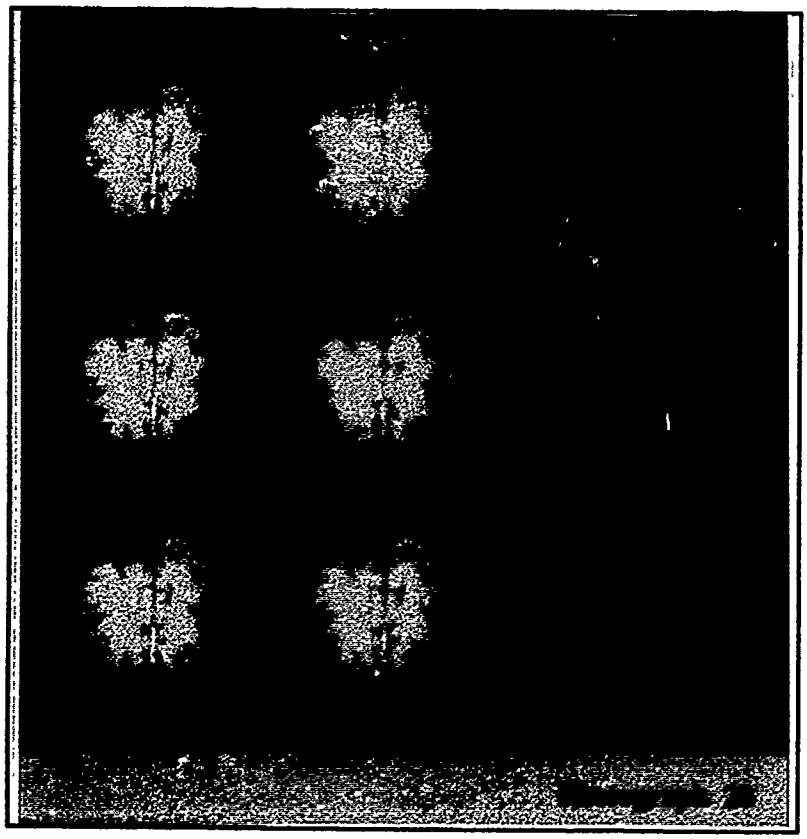


Figure C.20. Slice 18, Semi-Inclusion Distance = 5, 4, 3

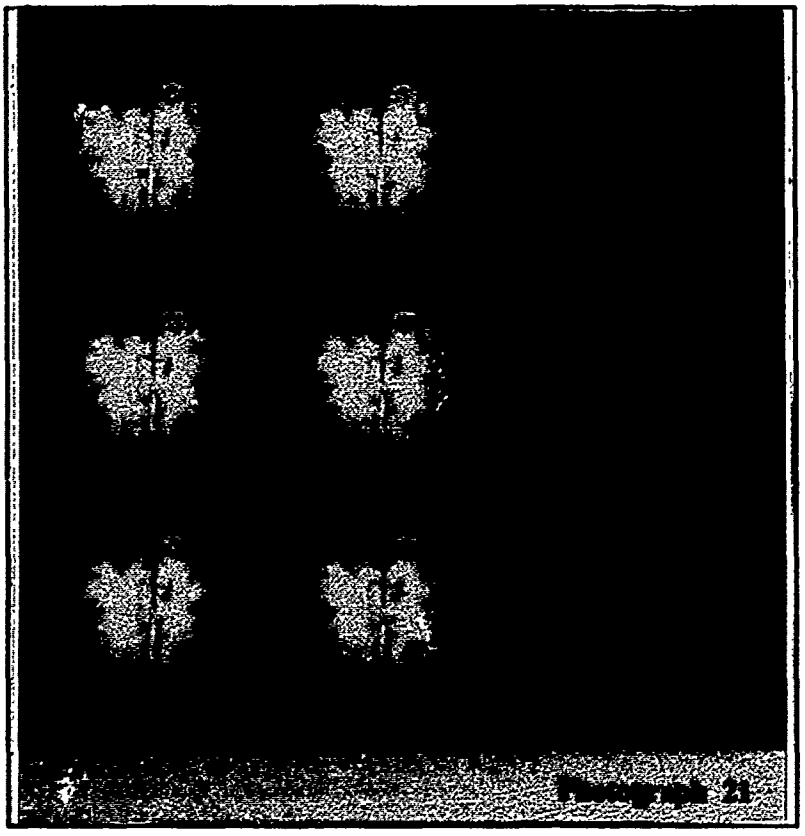


Figure C.21. Slice 18, Semi-Inclusion Distance = 3, 2, 1

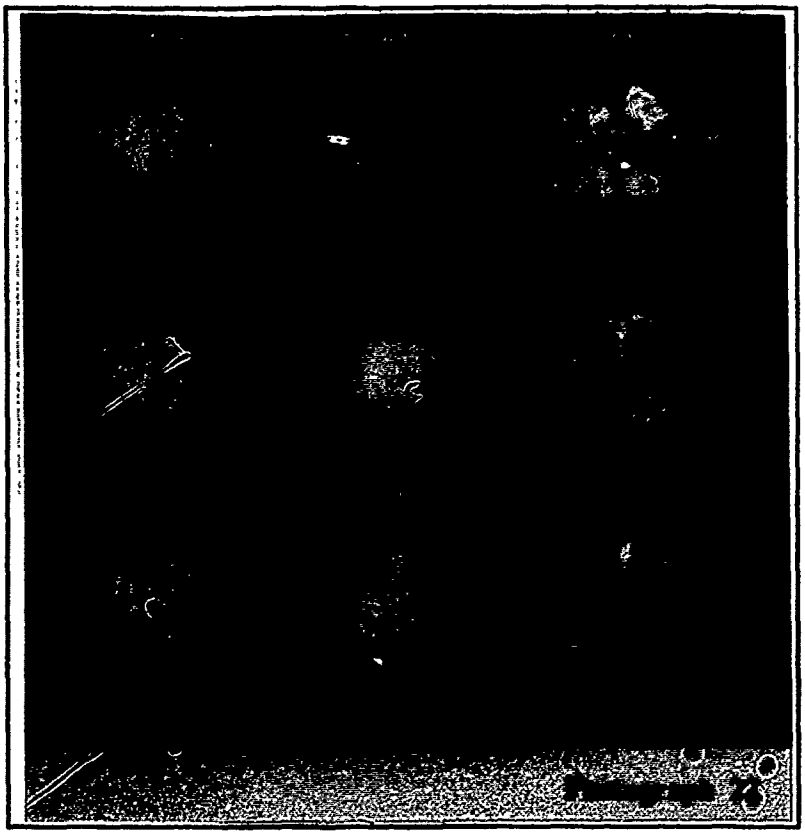


Figure C.22. Slice 21, Maximum Variance = 100, 85, 70

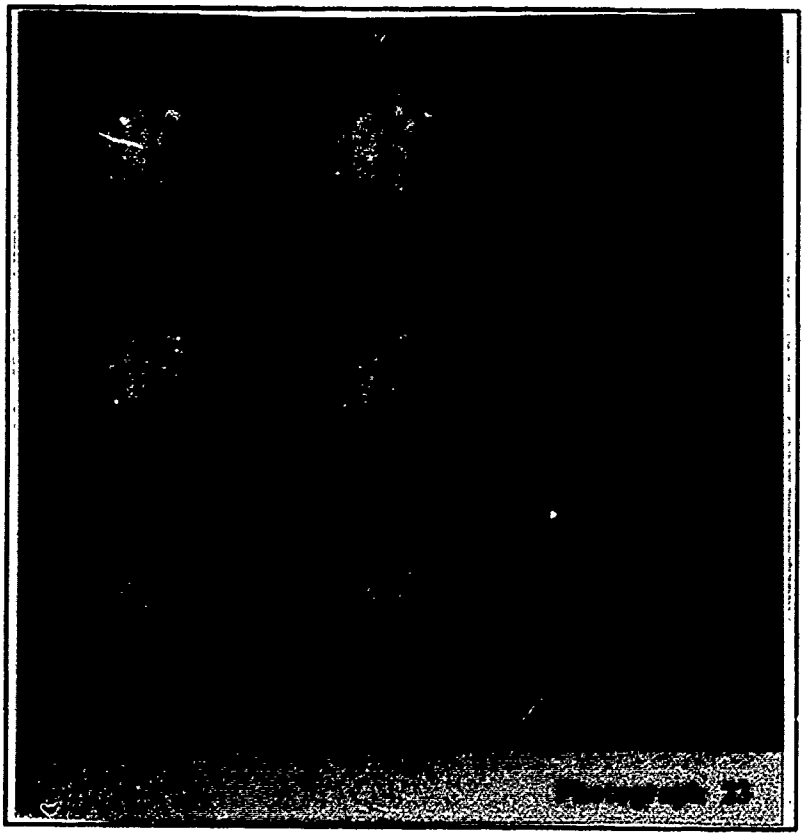


Figure C.23. Slice 21, Semi-Inclusion Distance = 5, 4, 3

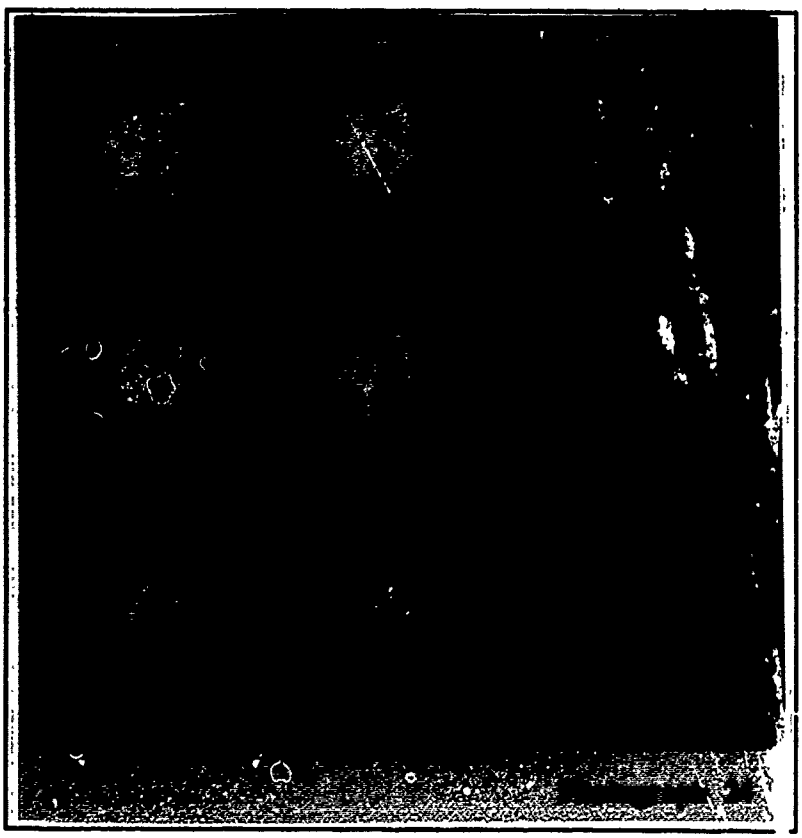


Figure C.24. Slice 21, Semi-Inclusion Distance = 3, 2, '

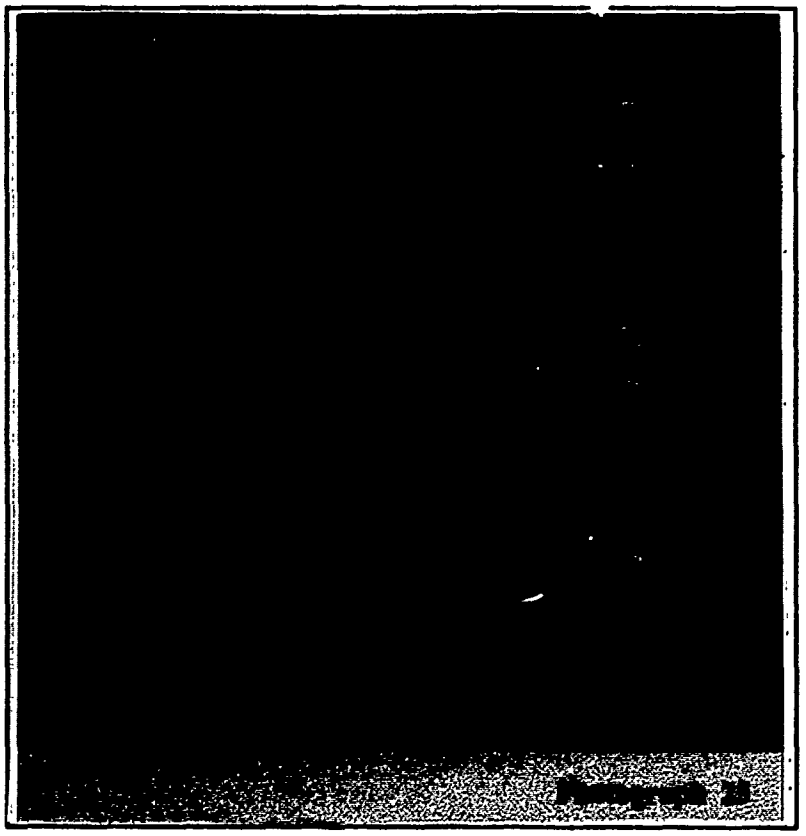


Figure C.25. Slice 24, Maximum Variance = 100, 85, 70

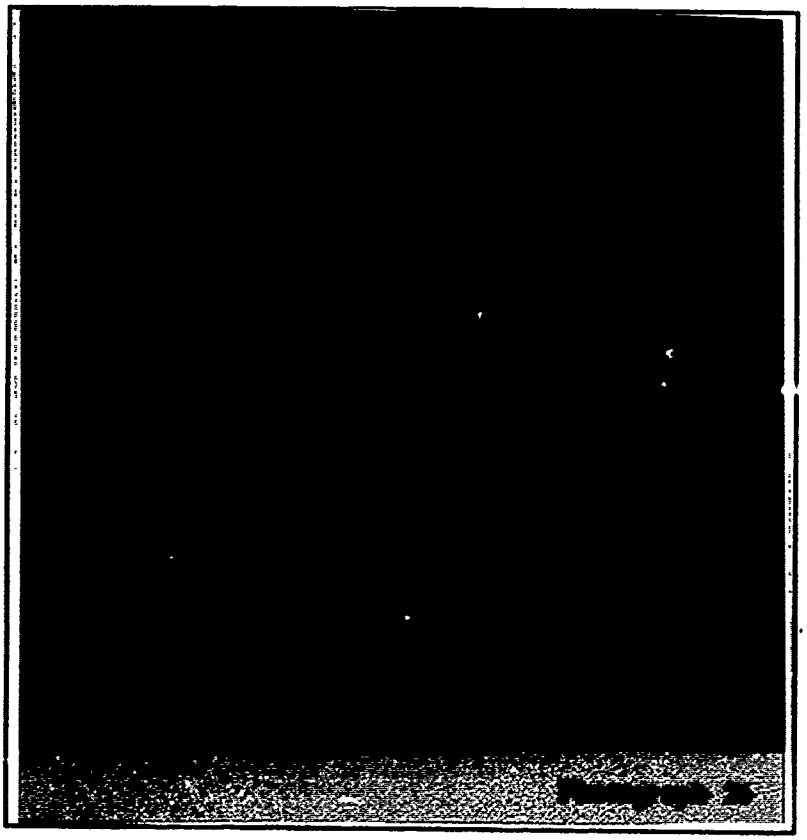


Figure C.26. Slice 24. Semi-Inclusion Distance = 5, 4, 3

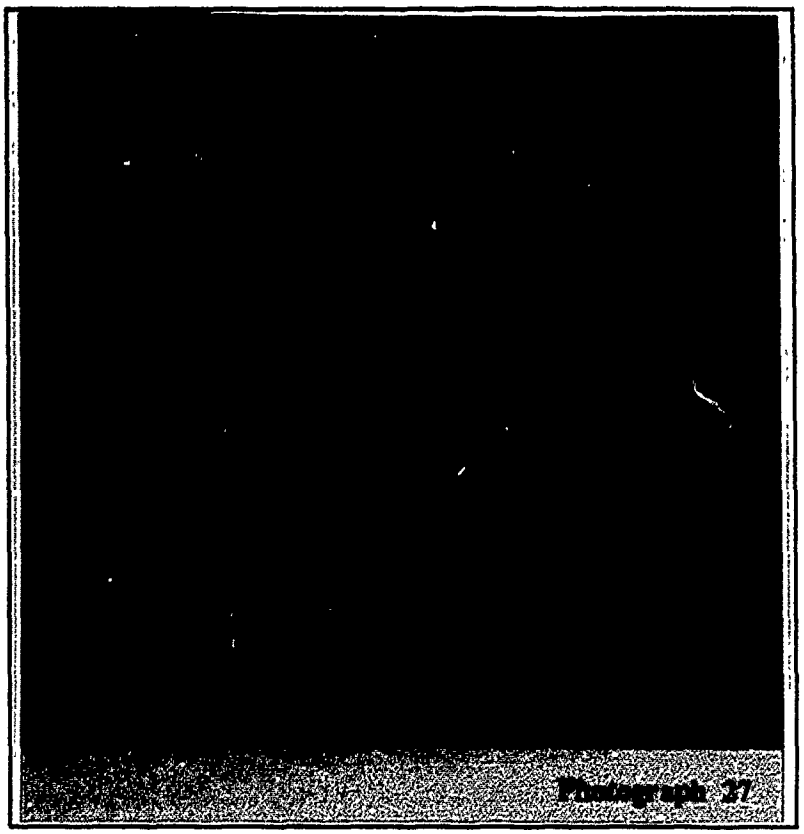


Figure C.27. Slice 24, Semi-Inclusion Distance = 3, 2, 1

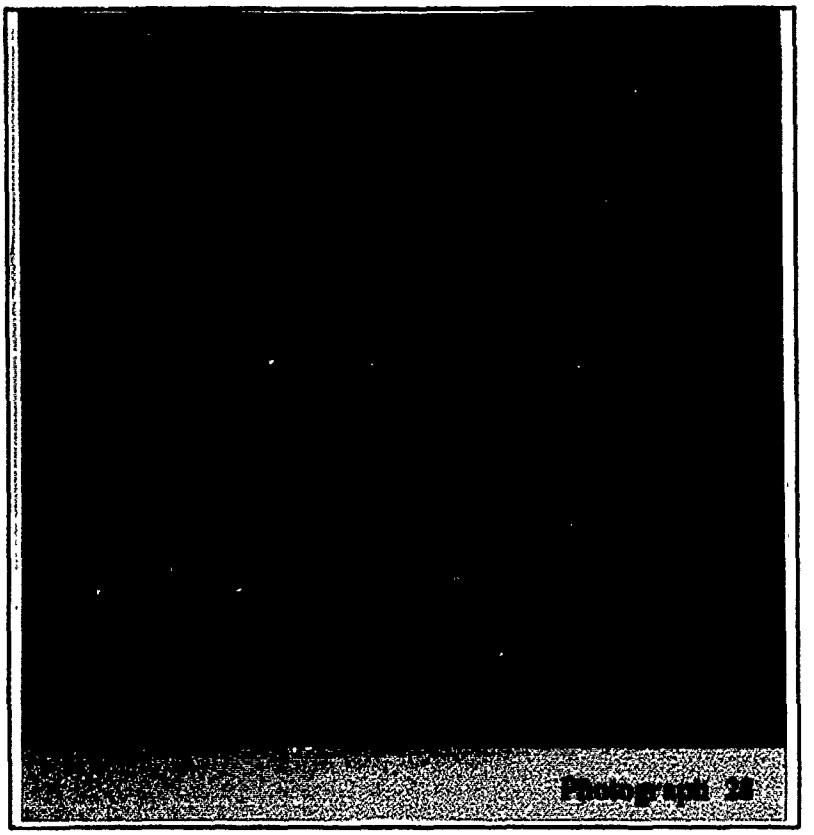


Figure C.28. Slice 24, Partitioned, Maximum Variance = 13, 12, 11

# Appendix D. *Minimized Image Tables*

This appendix includes the kriged image statistics for each slice of MRI data used in the kriging procedures. The run times were for all jobs running concurrently.

Table D.1. Slice 1 Statistics

| Minimization Method | | Largest Variance | Percent Reduction | Run Time hr:min:s | Photograph |
|---|---|---|---|---|---|
| Maximum Allowed Variance | 100 | 98.98 | 98.70 | 00:49:31 | 1(a) |
| | 85 | 84.39 | 96.53 | 00:38:15 | 1(b) |
| | 70 | 68.37 | 74.90 | 00:54:45 | 1(c) |
| Initial Semi-Inclusion Distances IAX, IAY | 5, 5 | 91.56 | 97.88 | 00:07:26 | 2(a) |
| | 4, 4 | 85.72 | 96.77 | 00:05:20 | 2(b) |
| | 3, 3 | 79.72 | 94.34 | 00:04:48 | 2(c), 3(a) |
| | 2, 2 | 73.50 | 87.36 | 00:06:57 | 3(b) |
| | 1, 1 | 66.92 | 50.00 | 00:13:58 | 3(c) |

Table D.2. Slice 3 Statistics

| Minimization Method | | Largest Variance | Percent Reduction | Run Time hr:min:s | Photograph |
|---|---|---|---|---|---|
| Maximum Allowed Variance | 100 | 99.42 | 99.39 | 00:54:48 | 4(a) |
| | 85 | 83.44 | 98.91 | 00:39:52 | 4(b) |
| | 70 | 69.74 | 97.80 | 00:37:23 | 4(c) |
| Initial Semi-Inclusion Distances IAX, IAY | 5, 5 | 70.27 | 97.88 | 00:03:40 | 5(a) |
| | 4, 4 | 63.73 | 96.77 | 00:03:42 | 5(b) |
| | 3, 3 | 57.09 | 94.34 | 00:04:06 | 5(c), 6(a) |
| | 2, 2 | 50.24 | 87.36 | 00:04:36 | 6(b) |
| | 1, 1 | 42.99 | 50.00 | 00:10:40 | 6(c) |

Table D.3. Slice 6 Statistics

| Minimization Method | | Largest Variance | Percent Reduction | Run Time hr:min:s | Photograph |
|---|---|---|---|---|---|
| Maximum Allowed Variance | 100 | 96.82 | 99.38 | 00:54:26 | 7(a) |
| | 85 | 84.16 | 98.94 | 00:42:16 | 7(b) |
| | 70 | 69.17 | 97.10 | 00:37:51 | 7(c) |
| Initial Semi-Inclusion Distances IAX, IAY | 5, 5 | 73.46 | 97.88 | 00:05:41 | 8(a) |
| | 4, 4 | 68.12 | 96.77 | 00:06:04 | 8(b) |
| | 3, 3 | 62.64 | 94.34 | 00:08:27 | 8(c), 9(a) |
| | 2, 2 | 56.98 | 87.36 | 00:05:41 | 9(b) |
| | 1, 1 | 50.97 | 50.00 | 00:16:25 | 9(c) |

Table D.4. Slice 9 Statistics

| Minimization Method | | Largest Variance | Percent Reduction | Run Time hr:min:s | Photograph |
|---|---|---|---|---|---|
| Maximum Allowed Variance | 100 | 98.69 | 98.33 | 00:45:04 | 10(a) |
| | 85 | 82.09 | 93.63 | 00:39:55 | 10(b) |
| | 70 | 69.18 | 25.09 | 01:34:58 | 10(c) |
| Initial Semi-Inclusion Distances IAX, IAY | 5, 5 | 95.44 | 97.88 | 00:03:33 | 11(a) |
| | 4, 4 | 89.79 | 96.77 | 00:03:46 | 11(b) |
| | 3, 3 | 83.99 | 94.34 | 00:04:42 | 11(c),12(a) |
| | 2, 2 | 77.97 | 87.36 | 00:05:51 | 12(b) |
| | 1, 1 | 71.61 | 50.00 | 00:19:48 | 12(c) |

Table D.5. Slice 12 Statistics

| Minimization Method | | Largest Variance | Percent Reduction | Run Time hr:min:s | Photograph |
|---|---|---|---|---|---|
| Maximum Allowed Variance | 100 | 97.13 | 99.60 | 00:55:45 | 13(a) |
| | 85 | 84.39 | 99.32 | 00:42:42 | 13(b) |
| | 70 | 69.52 | 98.67 | 00:38:55 | 13(c) |
| Initial Semi-Inclusion Distances IAX, IAY | 5, 5 | 62.49 | 97.88 | 00:03:57 | 14(a) |
| | 4, 4 | 57.16 | 96.77 | 00:04:18 | 14(b) |
| | 3, 3 | 51.72 | 94.34 | 00:04:41 | 14(c),15(a) |
| | 2, 2 | 46.10 | 87.36 | 00:06:01 | 15(b) |
| | 1, 1 | 40.13 | 50.00 | 00:19:41 | 15(c) |

Table D.6. Slice 15 Statistics

| Minimization Method | | Largest Variance | Percent Reduction | Run Time hr:min:s | Photograph |
|---|---|---|---|---|---|
| Maximum Allowed Variance | 100 | 98.92 | 99.84 | 01:17:16 | 16(a) |
| | 85 | 83.58 | 99.78 | 00:58:32 | 16(b) |
| | 70 | 70.00 | 99.70 | 00:30:11 | 16(c) |
| Initial Semi-Inclusion Distances IAX, IAY | 5, 5 | 26.64 | 97.88 | 00:04:07 | 17(a) |
| | 4, 4 | 21.90 | 96.77 | 00:05:07 | 17(b) |
| | 3, 3 | 17.16 | 94.34 | 00:05:42 | 17(c),18(a) |
| | 2, 2 | 12.42 | 87.36 | 00:09:26 | 18(b) |
| | 1, 1 | 7.65 | 50.00 | 00:34:58 | 18(c) |

Table D.7. Slice 18 Statistics

| Minimization Method | | Largest Variance | Percent Reduction | Run Time hr:min:s | Photograph |
|---|---|---|---|---|---|
| Maximum Allowed Variance | 100 | 98.20 | 99.91 | 02:08:07 | 19(a) |
| | 85 | 84.64 | 99.90 | 01:05:45 | 19(b) |
| | 70 | 69.81 | 99.84 | 00:34:25 | 19(c) |
| Initial Semi-Inclusion Distances IAX, IAY | 5, 5 | 17.12 | 97.88 | 00:03:49 | 20(a) |
| | 4, 4 | 13.69 | 96.77 | 00:04:09 | 20(b) |
| | 3, 3 | 10.27 | 94.34 | 00:04:38 | 20(c),21(a) |
| | 2, 2 | 6.84 | 87.36 | 00:06:04 | 21(b) |
| | 1, 1 | 3.42 | 50.00 | 00:15:32 | 21(c) |

Table D.8. Slice 21 Statistics

| Minimization Method | | Largest Variance | Percent Reduction | Run Time hr:min:s | Photograph |
|---|---|---|---|---|---|
| Maximum Allowed Variance | 100 | 99.92 | 99.96 | 01:36:19 | 22(a) |
| | 85 | 84.67 | 99.95 | 01:39:50 | 22(b) |
| | 70 | 69.45 | 99.93 | 01:27:51 | 22(c) |
| Initial Semi-Inclusion Distances IAX, IAY | 5, 5 | 9.17 | 97.88 | 00:04:07 | 23(a) |
| | 4, 4 | 7.34 | 96.77 | 00:04:48 | 23(b) |
| | 3, 3 | 5.50 | 94.34 | 00:05:40 | 23(c),24(a) |
| | 2, 2 | 3.67 | 87.36 | 00:09:25 | 24(b) |
| | 1, 1 | 1.83 | 50.00 | 00:34:57 | 24(c) |

Table D.9. Slice 24 Statistics

| Minimization Method | | Largest Variance | Percent Reduction | Run Time hr:min:s | Photograph |
|---|---|---|---|---|---|
| Maximum Allowed Variance | 100 | 53.15 | 99.96 | 01:42:16 | 25(a) |
| | 85 | 53.15 | 99.96 | 02:02:15 | 25(b) |
| | 70 | 53.15 | 99.96 | 01:41:20 | 25(c) |
| Initial Semi-Inclusion Distances IAX, IAY | 5, 5 | 3.94 | 97.88 | 00:04:06 | 26(a) |
| | 4, 4 | 3.15 | 96.77 | 00:04:25 | 26(b) |
| | 3, 3 | 2.36 | 94.34 | 00:05:18 | 26(c),27(a) |
| | 2, 2 | 1.57 | 87.36 | 00:07:36 | 27(b) |
| | 1, 1 | 0.79 | 50.00 | 00:20:29 | 27(c) |

Table D.10. Slice 24, Partitioned, Statistics

| Minimization Method | | Largest Variance | Percent Reduction | Run Time hr:min:s | Photograph |
|---|---|---|---|---|---|
| Maximum Allowed Variance | 13 | 12.45 | 96.65 | 00:08:00 | 28(a) |
| | 12 | 11.83 | 95.73 | 00:08:23 | 28(b) |
| | 11 | 10.77 | 91.04 | 00:10:58 | 28(c) |

# Appendix E. *Programmers Manual*

This appendix provides a description of the implementation of the routines needed for each program with explanations of how and why the coding is structured as presented.

## *E.1 Matrix Object*

To minimize a data set, the kriging program requires a matrix that can expand and contract as points are added to and deleted from the minimal data set. Typical matrix operations must also be supported. This can not be done using typical programming techniques. What is needed is an object oriented language. The matrix object is, therefore, written in C++ and supports the following requirements:

- Create any size matrix,
- Delete a matrix and free memory,
- Set an element of the matrix to a desired value,
- Get the value of an element of the matrix,
- Add a row at the bottom of the matrix,
- Add a column at the right edge of the matrix,
- Delete a row from the bottom of the matrix,
- Delete a column from the right edge of the matrix,
- Transpose the matrix,
- Invert the matrix,
- Reinvert the matrix after adding rows and columns,
- Premultiply the matrix by another matrix,
- Postmultiply the matrix by another matrix,
- Multiply the matrix by a scalar value,
- Add another matrix to this matrix,
- Copy one matrix to another,
- Return the number of rows of the matrix,
- Return the number of columns of the matrix,

- Test if last operation was successful.

Implementation of most of these routines is straight forward and will not be discussed. Two additional routines, *LUDCMP* (renamed *LU_Decomposition*) and *LUBKSB* (renamed *LU_Backsubstitute*), were adapted from *Numerical Recipes in C* (16). *LU_Decomposition* decomposes the matrix into the product of two matrices, $[L]$ and $[U]$, where $[L]$ is lower triangular and $[U]$ is upper triangular. *LU_Backsubstitute* performs back substitution on the decomposed matrix. The combination of these programs provided an efficient method for inverting the matrix. Of primary interest is the implementation of the object such that it has the ability to expand and contract.

The matrix object is implemented as a structure containing information about the matrix, such as number of rows and number of columns, and a pointer to an array of arrays in which the values are stored. The structure is illustrated in Figure E.1 which was adapted from the matrix object source code file. Each individual array, of the array of arrays, is sized as the minimum of 100 or the number of rows requested by the user for the row size of the array, and as the minimum of 100 or the number of columns requested by the user for the column size of the array. This implementation represents a compromise between excessive unused, but allocated, memory and excessive requests for memory with more overhead per value stored. By allocating large amounts of memory per array, the number of requests for memory and the administrative overhead keeping track of those allocations is minimized. But the amount of unused, but allocated, memory becomes excessive and, if carried to the extreme, can require more memory than is available. By allocating only enough memory for each row or column, unused memory is minimized, but, the number of requests for memory and the administrative overhead keeping track of all those allocations becomes excessive. Using the structure previously outlined, rows and columns can be added by allocating more arrays of memory. The number of rows and columns can be reduced very easily by adjusting the matrix information kept in

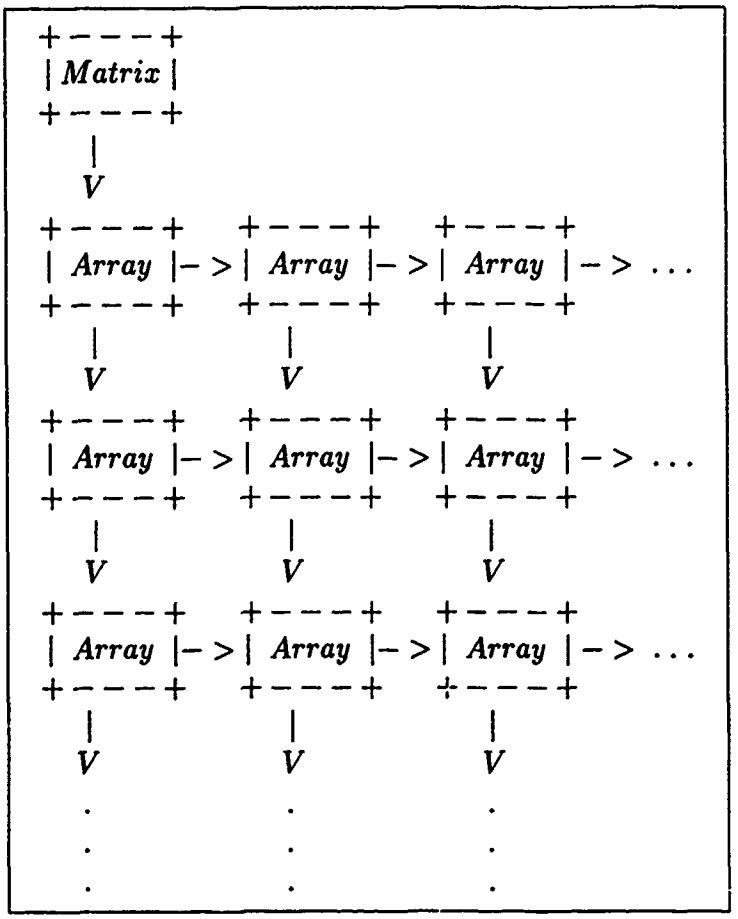



Figure E.1. Matrix Object Structure

the matrix structure. This does not return memory to the system but does appear, to the user, as if the rows and columns are deleted. Of several structures investigated this one appeared to be the closest to optimal. The matrix object is also used in the partitioning, residuals removal, and semivariogram estimating programs.

Of secondary interest is the implementation of the ReInvert routine. This routine uses Szidarovszky's matrix partitioning method to reinvert the matrix (19:192). Given the two matrices, $[A]$ and $[B]$, partitioned into submatrices as follows:

$$\left[ A \right] = \begin{bmatrix} P & Q \\ R & S \end{bmatrix} \quad \left[ B \right] = \begin{bmatrix} Y & Z \\ U & V \end{bmatrix}$$

where $[A] \cdot [B] = [I]$ and $P^{-1}$ is known. Performing the multiplication it can be seen that:

$$P \cdot Y + Q \cdot U = I \tag{E.1}$$

$$P \cdot Z + Q \cdot V = 0 \tag{E.2}$$

$$R \cdot Y + S \cdot U = 0 \tag{E.3}$$

$$R \cdot Z + S \cdot V = I \tag{E.4}$$

solving the second equation for $Z$:

$$Z = -P^{-1} \cdot Q \cdot V$$

combining this with the fourth equation and solving for $V$:

$$V = (S - R \cdot P^{-1} \cdot Q)^{-1}$$

solving the first equation for $Y$:

$$Y = P^{-1} \cdot (I - Q \cdot U) = P^{-1} - (P^{-1} \cdot Q) \cdot U$$

combining this with the third equation yields:

$$U = -(S - R \cdot P^{-1} \cdot Q)^{-1} \cdot R \cdot P^{-1} = -V \cdot R \cdot P^{-1}$$

therefore, the following relationships are established:

$$V = (S - R \cdot P^{-1} \cdot Q)^{-1}$$

$$Z = -P^{-1} \cdot Q \cdot V$$

$$U = -V \cdot R \cdot P^{-1}$$

$$Y = P^{-1} - (P^{-1} \cdot Q) \cdot U$$

If the number of added rows and columns is small relative to the original matrix size, the computational time saved by not inverting the entire matrix will be substantial.

### *E.2 Interface Routines*

In support of the kriging procedure's need for a common interface that isolates the input data format and control information format from the kriging and structural analysis procedures, an interface routines package, written in C++, is provided. The following are the routines provided for that standardized interface.

*Input_Check.* The number of command line arguments is tested to determine if a control information file name is provided and print usage information if not.

*Get_Data_Point.* Reads the data from the disk and assigns each point to a grid location. Returns row and column indices of the grid location as well as the coordinates of the point and the quantity of interest.

*Put_Data_Point.* Writes the data to the output file in the desired format. Row and column indices of the grid location as well as the coordinates of the point and the quantity of interest are sent as parameters.

*Output_Point.* Writes the data to the minimal data file in the desired format. Row and column indices of the grid location as well as the coordinates of the point and the quantity of interest are sent as parameters.

*Put_Variance.* Writes the estimation variance to the variance file in the desired format. Row and column indices of the grid location as well as the coordinates of the point and the variance are sent as parameters.

*Get_Control_Input.* This routine gets the program control information and opens the data files. The following control parameters can be set by the user:

- Number of increments on the $x$ and $y$ axes,
- Maximum values of $x$ and $y$,

- Minimum values of $x$ and $y$,
- Step size for $x$ and $y$,
- Zone of influence (range), a, in $0^0$ and $90^0$ directions,
- Sill minus the nugget effect, C, in $0^0$ and $90^0$ directions,
- Nugget effect, C0, in 0 and 90 degree directions,
- Angles at which to generate semivariograms,
- Correlations generated by semivariogram program for each theoretical model,
- The semi-inclusion distance for estimation or minimizing,
- Maximum allowable variance, used for data minimization ,
- Confidence interval to use calculating maximum allowable variance,
- Largest allowable difference of estimate from actual value,
- Minimum distance between points to consider them distinct,
- Minimization flag,
- Integer data flag,
- Input data file name,
- Output file name,
- Variance file name,
- Minimal data set file name,
- Plot file name,
- Semivariogram file name,
- Gridded data flag,
- Pattern output flag,
- Maximum number of points allowed in $[A]$ matrix,
- Polynomial coefficients,

- Semivariogram inclusion angle width,
- Semivariogram step,
- Total number of data points,
- Maximum lag for semivariogram calculation,
- Image expansion factor,
- Output inversion flag, and
- Number of header lines in data file.

The parameter input names, usage, and the actions taken based on those parameters are covered in detail in Appendix A, The User's Manual.

### *E.3 Kriging Routines*

To perform the tasks required by this thesis effort and the two concurrent thesis efforts, a kriging program originally written in C by Michael Grant (9) was rewritten in C++ with extensive modifications. The modifications correct some programmatic errors as well as incorporate geometric anisotropy, minimal data set production, and generalization to any data set. The program is long and somewhat complex but is composed of short and easy to understand routines. Therefore, to enhance understandability, each routine developed will be presented separately. Only those routines that directly support kriging will be described in this section. Those routines that support minimization will be presented in the minimization section. Those routines that have elements of both kriging and minimization will be presented in both sections. Note that a point is "kriged" if the estimate and the estimation variance at that point are calculated.

*Executive Routine.* The primary use of the kriging equations is the calculation of the estimate and the estimation variance at a point. This involves solving for the weights, $\{X\}$, from the kriging system of equations $[A] \cdot \{X\} = \{B\}$. To build the

matrices, the known points in the vicinity of the point being kriged must be found. Once the weights are known the point can be kriged. Therefore, the heart of the kriging program is the following routine calls:

- Get_Pts(i, j)
- Build_A()
- Build_B(i, j)
- Build_X()
- Estimate(i, j, Minimization_Flag)

For this effort, the points to be estimated lie on a regularly spaced grid. This allows the coordinates of a point to be specified by the indices, $i$ and $j$, of a two dimensional array. The routines that need to know the coordinates of the point being kriged are passed the array indices from which the coordinates can then be calculated. This sequence needs some modification, however, for this application. First, a point without known points in the neighborhood on which to base an estimate or a point that is all ready known can not be kriged. Therefore, *Get_Pts* will return zero if the point is known or if there are no points in the neighborhood. If minimization is being performed, the presence of a point at the grid location can be determined directly. The variable *Unknown_Point* is set to true if the point is to be kriged. The other routine calls are then placed within a block *if* statement testing on *Unknown_Point*. A second modification involves the routine to build the $[A]$ matrix. This routine is called only if minimization is NOT being performed. This will be discussed in the minimization section.

The kriging routines need to be exercised for all the grid locations. This is accomplished with two nested *for* loops. The indices of the *for* loops are variables so that subsections of the entire grid may be kriged. This will be very useful for minimization as will be explained later. Minimization also requires accomplishing

the doubly nested *for* loops after a known data point is added to or removed from the minimal data set. Therefore, these two loops need to be nested within a construct that is executed at least once for ordinary, non-minimizing, kriging. The construct used is the *do while* loop.

Before any kriging can be accomplished some program set-up is required. First, program control information must be acquired using the *Get_Control_Input* routine. Second, the data must be read in from the disk using the *Data_In* routine. Lastly, data structures must be allocated and variables initialized. *Initialize_Values* sets variables used for minimization.

After the data is kriged it is written to the disk with the *Output* routine. The pattern of known points may also be printed. Within the nested loops is a print section to write to a file named "*Status.Report*" the current location in the data field and some other values of interest.

*Data_In.* The construct used to hold the data points is an array of pointers to a structure that contains the coordinates of the point and the value of interest as well as a pointer to the next point collocated at this grid location. This is, in essence, an array of bins in which to place the data. This construct is used since many points from an irregularly spaced data field may be assigned to the same grid location. For minimization two of these constructs are used, one to contain all the data and one to contain the minimal data set.

To get the data from disk a *while* loop is used. The data is read from the disk and assigned to a grid location by the interface routine *Get_Data_Point*. When all the data is read in *Get_Data_Point* returns the End Of File (EOF) value and terminates the *while* loop. A small amount of memory is allocated for each datum read in and kept. Not all data will be kept. Duplicate data points are removed because they create an equation, in the kriging system of equations, that is not linearly independent of the others. If a matrix contains some equations that are linear combinations of others in the matrix, the matrix is singular and can not be

inverted. If the matrix can't be inverted the weights can't be calculated and the point can't be estimated.

*Get_Pts.* This routine is in two major sections: get the points at the grid location being kriged and get the points within a square neighborhood centered around the point being kriged. The square must be as large as possible subject to the following two constraints: 1) the total number of points must not exceed the number at which the matrix inversion routine becomes unreliable; and, 2) the size of the square must not exceed the size set by the user. To accomplish this, points are added from a square that is only one row and column larger above, below, to the left, and to the right than the last square. This is repeated until the point limit or square size limit is reached. This then reduces to adding the points from the edges of an "expanding" square, centered around the point being kriged, that starts with three rows and three columns and ends when one of the limit conditions is reached.

The first section is a *while* loop that gets all the points at the current grid location. Within the loop is a test to determine if one of the points coincides with the grid location. If so, that point is used and kriging is not performed. The routine returns zero in this case.

The second section is an outer *for* loop that increments the distance from the central point to the edges of the expanding square. Within this loop are four loops to get the points from each edge of the square. These four loops are very similar with only minor differences to ensure that the same point is not sampled twice. Therefore, only one of the loops will be described. Each loop starts by calculating the range for the indices of the *for* loop. The range is tested to make sure it does not fall outside the grid. The range is also tested to make sure that the points on that edge have not all ready been sampled. A loop almost identical to that in the first section is then performed on each location on that edge. The differences between this loop and the first are: points are not tested to determine if they coincide with the point being kriged and the distance from the point being kriged is calculated so that the

point will be included only if it is within the theoretical semivariogram range.

The points are placed in an expanding matrix. This is done so that the number of points to use for kriging can be set by setting the variable *Kpts*. The number of points added to the matrix upon completion of the loop to add points from the last edge of the previous square is saved in the variable *Last_Kpts*. If the number of points added to the matrix from the current edge of the expanding square exceeds the maximum set by the user then the number of points to use for kriging is set to *Last_Kpts*. There is a maximum of approximately 250 points that the current matrix inversion routine can reliably handle.

*Build_A.* This routine is in five sections and is relatively easy to follow. The first section resizes $[A]$ to the necessary dimensions. The second section uses a doubly nested *for* loop to cycle through the sample points matrix and determine the covariance between each pair of known points and then put that value into the $[A]$ matrix. The *Sigma* routine calculates the covariance based on the distance between the points. The third section adds a row and column of ones so that the weights will sum to one. The fourth section adds the terms to remove the local drift. The drift is assumed linear. The terms are calculated based upon a relative origin located at the upper left corner of the area being kriged. This is necessary for minimization and will be explained in the minimization section. The fifth, and last, section sets the block of zeros required by the matrix form of the kriging system of equations. After the matrix is constructed the inverse is calculated.

*Build_B.* This routine is in three sections and is also easy to follow. The first section resizes $\{B\}$ to the necessary dimensions. The second section uses a *for* loop to cycle through the sample points matrix and determine the covariance between each known point and the point to be kriged and then put that value into the $\{B\}$ matrix. The *Sigma* routine calculates the covariance based on the distance between the points. The third section puts the one for the sum of the weights and the values for the local drift into the matrix. The values for the local drift are calculated based

on the relative origin described in the previous section.

*Build_X.* The $\{B\}$ matrix is copied into the $\{X\}$ matrix and the $\{X\}$ matrix is then premultiplied by the $[A]^{-1}$ matrix. There is more coding to minimize computational time if minimization is being performed and will be discussed under the minimization section.

*Estimate.* The estimate and estimation variance is calculated using the equations presented in the literature review and stored into the surface and variance matrices. If the semivariogram is used in place of the covariance in the *Sigma* routine the formula for the estimation variance must be changed to the semivariogram form.

*Output.* The surface estimates, estimation variances, and, if produced, the minimal data set are written to the disk using the appropriate interface routine. Doubly nested *for* loops are used to cycle through the matrices. There is some extra coding to reverse the output format of the data if the user so desires.

*Sigma.* The covariance for the spherical theoretical semivariogram model is calculated using the equation presented in the literature review that corrects for geometric anisotropy. An average sill is calculated in the executive routine and used in this routine. Zonal anisotropy is beyond the scope of this effort and is not accounted for in the covariance calculation. This can be changed to return the semivariogram without impacting any routine other than the *Estimate* routine.

*min0.* This routine returns the smaller of the two integer parameters.

The following routines are added to support minimization.

*Add_Points.* This routine adds points based on the semi-inclusion distances and fills the sample matrix with the data that is within the current kriging pattern. The surface value at each added point location is set to the value of the quantity of interest of that point and the estimation variance at that point is set to zero. All this is accomplished very simply with doubly nested *for* loops.

*Initialize_Values.* This routine adds the initial points to the minimal data set based on the semi-inclusion distances. If the user does not set the semi-inclusion distances they will be calculated, in the executive routine, based on the theoretical semivariogram range. The calculation is based on a square kriging area with the distance between the center point and a corner point being just less than the theoretical semivariogram range. The user may input the number of times to divide the semi-inclusion distance by two. This is useful if the semivariogram range is large and the maximum variance is small because it reduces the size of the kriging area to start with. This routine also determines if the entire matrix is to be minimized or if submatrices are to be minimized and sets the minimization flag appropriately.

*Test_Set_And_Add.* This routine is long but is in ten distinct and fairly independent sections. The flow of control is from top to bottom and the interaction is limited to consecutive sections. Which section is currently being executed is controlled by the variable *Minimization_Flag* and indicates which submatrix, or kriging area, is currently being minimized or kriged. *Minimization_Flag* takes on the following values and meanings:

```
-1 - minimizing entire matrix due to large range
 0 - minimizing square central submatrix
 1 - kriging central submatrices
 2 - kriging left edge submatrices
 3 - kriging top edge submatrices
 4 - kriging right edge submatrices
 5 - kriging bottom edge submatrices
 6 - kriging upper left corner submatrix
 7 - kriging upper right corner submatrix
 8 - kriging lower left corner submatrix
 9 - kriging lower right corner submatrix
```

Since a matrix is used to represent most of required data constructs the phraseology is coached in terms of matrices. These ten sections fall under two major tasks: add points to reduce the largest estimation variance and krige a subarea of the grid with the minimal data set. Therefore, these two major tasks will be addressed instead of the ten sections.

As each submatrix is kriged the location and value of the largest estimation variance is recorded. If the submatrix equal to the entire matrix is being minimized and the largest variance is greater than the maximum allowed variance a point is added at the recorded location and the matrix is rekriged. If a submatrix not equal to the entire matrix is being minimized and a point must be added then symetric points about an x and y axis through the center point of the kriging pattern must also be added as previously explained. If the kriging pattern is replicated and the semi-inclusion distance is divisible by two then the size of the kriging area will be reduced. If the maximum variance criteria is met then set-up for the next section must be performed. If the entire matrix is being minimized then minimization is complete as the whole array has all ready been kriged. If a submatrix is being minimized then the first kriging section must be initialized.

Each kriging section, except for the corner sections, must increment the row and column submatrix indices to the next submatrix to krige. If the first submatrix of the section is being kriged then the *Build_X* routine must be instructed to save the weights, otherwise *Build_X* must be instructed to use the previously saved weights. Every section must restock the sample array with the points from the current submatrix being kriged. Each section must also determine if all the submatrices in that section have been kriged and, if so, set-up the next section. The section kriging the central submatrices must also determine when the last submatrix of the current submatrix row is completed so that the column indices can be set to the first submatrix of the row and the row indices can be incremented to the next submatrix row. The coding of this routine should now be relatively easy to understand.

When minimization is finished the minimization flag is set to false to allow the *do while* loop in the executive routine to stop looping. The minimized flag is set to true so that the *Output* routine will write the minimal data set to the disk.

*ReBuild_A.* This routine adds the extra terms to the $[A]$ matrix required by the addition of extra points to the kriging area. It is structured very similarly to *Build_A*. The difference is the starting indices for the loops. *Build_A* fills the entire matrix whereas this routine adds to the matrix on the right and at the bottom. This was done so that the *ReInvert* method of the matrix object could be used to save some computational time.

*Build_A.* For $[A]^{-1}$ of the first kriged pattern to equal $[A]^{-1}$ of subsequently kriged identically patterned areas all terms in the $[A]$ matrix must be identical. This includes the values used to account for the local drift. For these values to be identical from pattern to pattern they must have the same relative origin. Therefore, the upper left corner of each kriging area is used as the relative origin for that area. This origin is also used in *ReBuild_A* and *Build_B*.

*Build_X.* There are two sections of code added for minimization. The first section uses the previously saved weights if the flag to use the saved weights is set otherwise it calculates the weights. The second section saves the weights if the flag to save the weights is set.

*Calculate_IAX_IAY.* This routine is implemented as a single *do while* loop that terminates when the estimation variance for the center point of a rectangular kriging pattern is smaller than the maximum allowed variance. There is an *if* structure to determine the correct number and coordinates of the kriging pattern based on whether the kriging pattern is to be used or if normal kriging is to be used. To reduce the distance between the points for each succesive pass through the loop a multiplier is used that is decremented on each pass through the loop.

# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188



| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | Dec 91 | Master's Thesis |

**4. TITLE AND SUBTITLE**
The Application of Kriging for Controlled Minimization of Large Data Sets

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**
Christopher D. Brodkin

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Air Force Institute of Technology
Wright-Patterson AFB, OH, 45433-6583

**8. PERFORMING ORGANIZATION REPORT NUMBER**
AFIT/GA/ENY/91D-14

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Jennifer Whitestone
AL/CFHD
Wright-Patterson AFB, OH, 45433

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release, distribution unlimited

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

This thesis develops the application of kriging for the controlled minimization of large data sets based on a maximum acceptable level of error. Specifically, the geostatistical estimation technique of kriging is used to produce minimal data sets and to estimate the unknown values on an arbitrarily sized grid using as input any data set. All the procedures necessary to improve the accuracy of the estimate as well as the kriging procedure aredeveloped. Using these procedures the entire process can be automated. The techniques are demonstrated using Magnetic Resonance Image data to support minimizing on-line storage requirements. Two concurrent thesis efforts also use the techniques to enlarge satellite photographs and to change the grid resolution of terrain data.

| 14. SUBJECT TERMS | 15. NUMBER OF PAGES |
|---|---|
| Statistical Functions, Covariance, Anisotropy, Interpolation, Structural Analysis, Variogram | 159 |
| | 16. PRICE CODE |

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UL |