

**Effect of Manual Digitizing Error on the Accuracy
and Precision of Polygon Area and Line Length**

by

Brenton Jan Keefer

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

APPROVED:

James L. Smith, Chairman

John Scrivani

Timothy G. Gregoffe

November 1988
Blacksburg, Virginia

**Effect of Manual Digitizing Error on the Accuracy
and Precision of Polygon Area and Line Length**

by

Brenton Jan Keefer

James L. Smith, Chairman

Forestry

(ABSTRACT)

Manual digitizing has been recognized by investigators as a significant source of map error in GIS, but the error characteristics have not been well defined. This thesis presents a methodology for simulating manual digitizing error. Stream mode digitizing error was modeled using autoregressive moving average (ARMA) procedures, and point mode digitizing was stochastically simulated using an uniform random model. These models were developed based on quantification of digitizing error collected from several operators. The resulting models were used to evaluate the effect digitizing error had upon polygon size and total line length at varying map accuracy standards.

Digitizing error produced no bias in polygon area. The standard deviation of polygon area doubled as the accuracy standard bandwidth doubled, but the standard deviation was always less than 1.6 percent of total area for stream mode digitizing. Smaller polygons (less than 10 square map inches) had more bias and more variance relative to their size than larger polygons. A doubling of the accuracy standard bandwidth caused a quadrupling of line length bias and a doubling to tripling of the line length standard deviation. For stream mode digitizing, reasonable digitizing standards produced line length biases of less than 2 percent of total length and standard deviations of less than 1 percent of total length. Bias and standard deviation both increased with increasing line length (or number of points), but the bias and standard deviation as a percent of total line length remained constant as feature size changed.

Acknowledgements

It has been my distinct privilege to have worked on this thesis with the members of my advisory committee, Dr. James L. Smith, Dr. Timothy Gregoire, and Dr. John Scrivani. Their combined expertise, knowledge, and advice made completing this thesis a more enjoyable and educating experience, and I extend my thanks to them. I am deeply grateful for the advice and support of Jim Smith, my major professor, who has served in the roles of both advisor and friend.

Most importantly, I owe my thanks to my wife, _____, who has faithfully supported me throughout both my undergraduate and graduate careers. Without her love, support, patience, and sacrifice, none of this would have been possible. Finally, I dedicate this thesis to my son, _____, in hope that he too will develop a love for learning and a desire to better understand the world around us.

Table of Contents

INTRODUCTION	1
LITERATURE REVIEW	3
MAP ACCURACY STANDARDS	3
GIS COMPONENTS	5
SOURCES OF MAP ERROR	10
MODELS OF MAP ERROR	12
Map Error in General	12
Digitizing Error	16
SUMMARY	18
MATERIALS AND METHODOLOGY	20
PHASE I: Measuring and Modeling	20
Data Collection	20
Measurement of Digitizing Error	22
Statistical Model	25
Test for Non-randomness	29

Model Specification and Fitting	30
PHASE II: Simulation	31
Overview of Simulation	31
Error Generation Process	32
PHASE III: Analysis	36
Data Collection	37
Analysis	37
RESULTS AND DISCUSSION	40
PHASE I: Measuring and Modeling	40
Empirical Data Analysis	40
Tests for Non-randomness	42
Model Specification and Fitting	50
PHASE II: Simulation	50
PHASE III: Analysis of Simulation	52
Error Magnitudes and Trends Across Accuracy Standards	52
Trends Across Feature Size	66
SUMMARY AND CONCLUSION	91
LITERATURE CITED	95
Appendix A. Test Features	99
Appendix B1. Error Measurement Program	102
Appendix B2. SIMDIGER	111

Appendix C. Empirical Data	124
Appendix D. Graphs of Simulation Results	127
VITA	180

List of Illustrations

Figure 1. Blakemore's epsilon bounded point in polygon check	13
Figure 2. Peucker's theory of the cartographic line	15
Figure 3. Perpendicular error distance between standard line and sample line	23
Figure 4. Intersection of vector A (standard line segment) and vector B (perpendicular error distance)	24
Figure 5. Possible locations of perpendicular intersections	26
Figure 6. Perpendicular intersection case 4 where perpendiculars can not be constructed	27
Figure 7. Distribution of stream mode digitizing error from five operators and 16 features; interval labels are uneven due to rounding error	43
Figure 8. Distribution of point mode digitizing error from five operators and 16 features; interval labels are uneven due to rounding error	44
Figure 9. Plot of stream mode digitizing error over digitizing interval	45
Figure 10. Plot of point mode digitizing error over digitizing interval	46
Figure 11. Distribution of correlation coefficients for stream mode digitizing error after fitting an AR(1) model	51
Figure 12. Average standard deviations of polygon area as percent of actual acres for a) stream mode, and b) point mode	58
Figure 13. Average bias of line length as percent of actual feet for a) stream mode, and b) point mode	61
Figure 14. Schematic of accuracy standards and error influenced line lengths	62
Figure 15. Average standard deviation of line length as a percent of actual feet for a) stream mode, and b) point mode	67
Figure 16. Bias of polygon area plotted against polygon area for stream mode simulation at alpha level .025	68
Figure 17. Relative bias of polygon area plotted against polygon area for stream mode	

	simulation at alpha level .025	70
Figure 18.	Relative bias of polygon area plotted against polygon area for point mode simulation at alpha level .025	71
Figure 19.	Standard deviation of polygon area plotted against polygon area for stream mode simulation at alpha level .025	73
Figure 20.	Relative standard deviation of polygon area plotted against polygon area for stream mode simulation at alpha level .025	75
Figure 21.	Relative standard deviation of polygon area plotted against polygon area for point mode simulation at alpha level .025	76
Figure 22.	Bias of line length plotted against line length for stream mode simulations at alpha level .025	78
Figure 23.	Bias of line length plotted against line length for point mode simulations at alpha level .025	79
Figure 24.	Bias of line length plotted against number of points per line for point mode simulations at alpha level .025	80
Figure 25.	Relative bias of line length plotted against line length for stream mode simulations at alpha level .025	82
Figure 26.	Relative bias of line length plotted against number of points per line for point mode simulations at alpha level .025	83
Figure 27.	Relative bias of line length plotted against points per foot for point mode simulations at alpha level .025	84
Figure 28.	Bias of line length plotted against points per foot for point mode simulations at alpha level .025	86
Figure 29.	Standard deviation of line length plotted against line length for stream mode simulations at alpha level .025	87
Figure 30.	Relative standard deviation of line length plotted against line length for stream mode simulations at alpha level .025	88
Figure 31.	Standard deviation of line length plotted against line length for point mode simulations at alpha level .025	90

List of Tables

Table 1.	Length of test feature for digitizing	21
Table 2.	Map accuracy standard parameters	38
Table 3.	Means and standard deviations of digitizing error over all features for each operator	41
Table 4.	Z-scores for runs tests for stream mode and point mode digitizing	48
Table 5.	Average bias of polygon area for stream mode simulations a) bias in actual acres, and b) bias in percent of actual acres	53
Table 6.	Average bias of polygon area for point mode simulations a) bias in actual acres, and b) bias in percent of actual acres	54
Table 7.	Average standard deviation of polygon area for stream mode simulations a) standard deviation in actual acres, and b) standard deviation in percent of actual acres	56
Table 8.	Average standard deviation of polygon area for point mode simulations a) standard deviation in actual acres, and b) standard deviation in percent of actual acres	57
Table 9.	Average bias of line length for stream mode simulations a) bias in actual feet, and b) bias in percent of actual feet	59
Table 10.	Average bias of line length for point mode simulations a) bias in actual feet, and b) bias in percent of actual feet	60
Table 11.	Average standard deviation of line length for stream mode simulations a) standard deviation in actual feet, and b) standard deviation in percent of actual feet	64
Table 12.	Average standard deviation of line length for point mode simulations a) standard deviation in actual feet, and b) standard deviation in percent of actual feet	65

INTRODUCTION

Computer cartography and geographic information systems (GIS) have developed into invaluable tools for the display and analysis of spatial information used in the decision making process of private companies and public agencies. Computer technology has advanced so rapidly that research of the automated map making process and research of the reliability of mapping systems have lagged behind. Many users of computer generated maps assume that the high quality appearance of the product implies high accuracy (Vitek and Richards, 1978). However, this is not always a valid assumption to make.

Components of the automated map making process introduce error into the system at many points. Contributions to error are made by source materials, computer equipment, software algorithms, and human deficiencies, just to name a few. Manual digitizing during the data input phase of computer cartography has been recognized to be a significant source of error, but it has not been well studied (Chrisman, 1982). Digitizing is usually the most expensive part of a GIS, yet the error introduced into maps by digitizing is often overlooked or assumed to be negligible. Although technological advances are increasing the use of automatic digitizing methods, manual digitizing is still the primary method of data capture used in the industry.

Manual digitizing is accomplished by tracing a map line using a hand held cursor. As the cursor moves along the line, the location of points are electronically sampled from the line and recorded as coordinates in some computer-compatible form. From these stored coordinates, the digitized line can be reproduced, and how well the line can be reproduced is a question of map accuracy. The human source of error in digitizing stems from a person's inability to follow a line perfectly. As the operator traces a line, he/she tends to displace the cursor from one side of the line to the other. This deviation affects how well the plotted computer map will duplicate the source map. Psychologists and cartographers have studied this phenomenon, but few quantitative models of the line-following process have been developed (Traylor, 1979).

Map accuracy is expressed and controlled by establishing map accuracy standards. Accuracy standards are often placed on the digitizing process in terms of a tolerance error band. However, within that stated band of error, the actual map line can assume many positions. Even if the accuracy standard can actually be adhered to by a digitizer, it conveys little useful information to a map user. The accuracy standard on the final map product should be in common map measurement units understood and used by the user (e.g. deviation of polygon area or line length). When a feature is digitized at the set accuracy standard and plotted, how that feature actually varies from the source map in terms of area or length is virtually unknown.

This study will concentrate on the effect digitizing error has upon map accuracy. There are two primary objectives: 1) to statistically model digitizing error, 2) to evaluate the effect of digitizing error upon polygon size and line length at varying map accuracy standards.

LITERATURE REVIEW

MAP ACCURACY STANDARDS

For centuries man produced and used hand-drawn, paper maps. Recently, computer generated maps have begun to replace the manually produced map. Whether map making is automated or manual, the accuracy of the map product has always been important, and unfortunately, it has often been a confusing issue to producer and user alike. The accuracy of a map is normally expressed as a map accuracy standard. Quite often a map user does not understand or is completely unaware of the accuracy level of the map he is using. Accuracy standards sometimes appear on maps, but the user usually can not apply the standard in the terms he desires (Vitek and Richards, 1978; Bennett, 1977).

The National Map Accuracy Standards (NMAS) approved in 1947 are the current standards used by USGS to produce paper topographic quad sheets (Marsden, 1960). The NMAS define standards for both horizontal positional accuracy and vertical positional accuracy. Large scale maps ($< 1:20000$) must have 90% of well defined points within $1/30$ of an inch of the true ground location. Small scale maps ($> 1:20000$) are to have 90% of well defined points within $1/50$ of an inch of true position. The vertical accuracy statement states that 90% of the

elevations must be within 1/2 the contour interval (Thompson, 1979). Several problems with the NMAS have been cited. The standards were not given in statistical terminology, and they were not easy to interpret literally for a particular map user's purpose. Also, the NMAS did not account for the distribution of the errors within the stated tolerance band (Thompson, 1960).

Various alternatives have been suggested for expressing map accuracy standards. The use of root mean squared error has been widely proposed as a statistical representation of accuracy (Thompson, 1979; Thompson and Rosenfield, 1971; Merchant, 1987). Instead of one set standard for all maps, several possible accuracy ratings could be given to a map depending on which criteria the map meets. This method was used in developing the Canadian Mapping Standards (Merchant, 1987; Surveys and mapping branch of Canadian map office, 1976). The American Society of Photogrammetry and Remote Sensing (ASPRS) began to revise the NMAS and include some of these alternatives (Merchant, 1987).

As automated mapping became established, it was unknown whether the NMAS could be applied to the new digital maps. Therefore, the USGS did not place quantified map accuracy statements on their digital topographic maps. They realized the accuracy was highly dependent upon source materials used to produce the digital model, but no actual accuracy statement could be made (Elassal and Caruso, 1983; Allder and Elassal, 1983).

As a result of growth in numerical and analytical cartography, the National Committee for Digital Cartographic Data Standards (NCDCCDS) was established to develop digital standards. NCDCCDS was set up under the auspices of the American Congress on Surveying and Mapping (ACSM) with the support of USGS (Moellering, 1982). The goal of the NCDCCDS was not so much to develop a strict set of standards, but to simply inform map users of the accuracy of the map they were using. This was dubbed a "truth in labeling" concept. The committee felt statements of map accuracy should be in terms familiar to the map user and in

terms of quantities obtained from the map by the user (eg. bounds on areas) (Chrisman, 1983b; Merchant, 1982). Most map users did not realize the limitations of their maps because map accuracy was poorly conveyed to the user. Statements of map accuracy normally were in terms not applicable to the user's objectives. Unfortunately, the saying that "nothing is quite so expensive as a cheap map" was quite true (Bennett, 1977).

One of the objectives of the committee was to establish standards for digitizing accuracy so as to control error in the map product caused by the digitizing process. Chrisman (1983a) emphasized the need for studying the variability introduced into maps as a result of digitizing methods. Another objective of the NCDCCDS was to develop methods for testing map accuracy. Often to test map accuracy, a computer generated map was simply laid over top of the original map and a comparison made. Major blunders were identified using this technique, but true accuracy standards were not determinable using this coarse method (Thompson, 1979; Thompson and Rosenfield, 1971). Graphical detection is very hard if the error present is less than two line widths (Chrisman, 1984). Vonderohe and Chrisman (1985) suggested testing digital maps against a source which is independent of the map making procedure. This source should be at an accuracy level three times that required by the product to be tested. This study tested maps by comparing point coordinates in digital form instead of using a graphical inspection.

GIS COMPONENTS

GIS and computer cartography in general are young and expanding disciplines. Computerized mapping came into being during the late 1950's and early 1960's with the development of SYMAP (SYnagraphic MAPping system) by Howard T. Fisher at Harvard University (Sheehand, 1979). By the end of the 1960's, SYMAP was operating at over 100 sites, and many

other mainframe-based systems were being developed such as GRID, IMGRID, and GEOMAP (Burrough, 1986). Map users were reluctant to accept computer generated maps at first due to the poor quality of the map products. Fortunately, technology advanced rapidly, and by the late 1970's, computer cartography had surpassed traditional cartographic methods in terms of product quality, product flexibility, and speed of delivery (Burrough, 1986).

As the field of computer cartography advanced, GIS emerged as a powerful tool which not only produced maps, but also provided for the analysis of spatial and attribute data. Prisley (1986) has defined a GIS to include the following: 1) digitization (computerization) of maps and associated data, 2) storage of digital data in organized fashion, 3) analysis of map-related data, and 4) production of maps, tables, and reports summarizing data.

Computer generated maps have many advantages over traditional paper based maps. Maps can be made at any specified scale. Maps with the same base features but varying themes are produced quickly and efficiently. Also, with the GIS capabilities, map analyses are available that would not have been possible with manual mapping methods. Drafting time of computer maps is also greatly reduced compared to manual drafting time (Cartsensen, 1987, personal communication).

Computer mapping and GIS became important decision making tools in many fields including forest resource management. The USDA Forest Service has been implementing and researching automated mapping and analysis for over 20 years. Several national forests have GIS in operation (Shumway, 1986). Many if not all major forest products firms incorporate some type of GIS into their management systems. Wakeley (1987), of the Weyerhaeuser Company, stated that GIS is "the most efficient and effective means of collecting, storing, and distributing the timberlands management information." GIS can provide decision making information in forest management activities from forest inventory to harvest scheduling to fire control.

Many individuals and institutions began research in the field of GIS. One of the leaders of research in GIS was the Harvard Graphics Laboratory which began producing the Harvard papers on GIS in 1978 (Fegeas, 1978). However, computer hardware and software advanced so rapidly that basic research regarding the computer mapping process and computer generated maps lagged behind. Peucker (1982) made the statement that "computer cartography is like a person with the body of an athlete in his prime time and the mind of a child." The subject of digital map accuracy in particular was not addressed satisfactorily. This lack of knowledge may be troubling to those who base important decisions on the products of GIS and automated mapping. Error is introduced into the map making process at many points, and one of the largest contributors to this error is the data input stage, which includes digitizing error. Unfortunately, very little research has been done on the subject of digitizing error (Traylor, 1979; Baugh and Boreham, 1976; Ottawa, 1987).

GIS store and process data using two basic structures, vector and/or raster. In a vector system, all features are stored as a series of points connected by straight lines. These points are described in the computer file by a Cartesian coordinate reference system. Conversely, in a raster storage system, a map is divided into a grid of cells, and a value representing each cell is stored in the data file. Raster systems are continuous in their method of storing data, as opposed to the discrete method of vector systems. In a raster system, features are not stored as individual entities as they are in vector mode. Manual digitizers were designed to convert map data into a vector formatted file, and automatic raster scanners converted source maps to raster format (Monmonier, 1982).

The automated mapping process can be divided into five main components. First, a data input stage is used to convert source map information into a computer-compatible digital form. This conversion is done using a manual digitizer or some type of automatic scanning device. Second, an editing stage corrects errors or makes adjustments to the data files. The third

stage is storage of the digital map and data files. Fourth, map analysis is used to integrate the spatial data with the attribute data if the mapping process is to include GIS capabilities. The fifth and final stage is outputting a computer generated map or some other type of report (Traylor, 1979).

The data input stage of computer cartography is often the most difficult and prone to error. Converting spatial data from graphic maps to digital form is "the first and most difficult step toward using the power of computer technology" (Fegeas, 1978). Data input can be categorized by four phases. First, the requirements for a particular mapping project must be determined and evaluated, as this will dictate the source maps to be used. The second phase is map acquisition and pre-processing where source maps are obtained, reformatted if necessary, and prepared for digitizing. Third, the maps are converted to digital format by digitizing. The fourth phase, initial editing, often is concurrent with digitizing. Obvious errors and blunders are corrected as the map is digitized (Marble et al., 1984).

Digitizing, the third phase of the data input stage, is simply the conversion of analog map data to digital map data (Blakeman, 1987). A cursor is manually traced over the map line, and coordinates are electronically sampled from the line. A digitizing table or tablet has thousands of tiny wires forming a grid under the table surface. Each intersection on the grid represents a coordinate. A paper map is placed on the table and as the cursor is moved along the map lines, the computer records the coordinates associated with a particular grid intersection.

Coordinates from the digitizer may be recorded either in stream mode or point mode. Stream mode means that as the cursor is moved along the map line, coordinates are recorded at some specified, regular time or distance interval. Many more points are recorded than are necessary to adequately define a line; consequently, some point thinning algorithm must be employed (Douglas and Peucker, 1973; Opheim, 1982). As a result of this line generalization,

some additional error is introduced into the computer map. Point mode allows the person digitizing to select when to record the coordinates of a point. He or she does so by pressing one of the buttons on the cursor. The digitizer must use judgment in knowing which points are needed to adequately characterize the line. Two types of points are often recorded: 1) points occupying significant economic, political, or cultural locations, and 2) points which are natural, important, or basic to the distinctive structure of the line (Jenks, 1981). Point mode is considered by some to be extremely tedious and to produce a poor representation of the true feature (Douglas and Peucker, 1973). Others promote point mode arguing that stream mode produces more error in the digitizing process (Burrough, 1986; Jenks, 1981). Stream mode is considered to be faster and easier for the digitizer. No actual studies have been conducted to examine the differences between digitizing in point mode or stream mode.

Manual digitizing began in the 1960's with mechanical-arm digitizers and rapidly advanced to the electromagnetic table and cursor of today. It was soon realized that manual digitizing was beset with problems. First, manual digitizing is a very slow process, and second, the operator induces error into the map because of human frailties such as physiological and psychological deficiencies (Shea, 1987). Attempts were made to remove the human factor from the digitizing process by automating digitizing. However, automatic digitizers also had many problems. Automatic raster scanners, the most successful and popular of the automated digitizing methods, recorded map data in a raster format. The raster format of the scanners did not interface with most computer mapping packages which used data manipulation processes that were vector oriented. Other disadvantages to raster data were the large storage requirement, and the high cost of the scanners (Cartsensen, 1987, personal communication).

Many of these problems still plague the acceptance of automatic digitization. However, in recent years, the cost of raster scanners has declined drastically. Raster format is becoming widely used in computer mapping and GIS programs, and it is possible to convert from

raster to vector or from vector to raster. New systems were developed which incorporated raster scanning with interactive line following and editing (Gossard, 1986). Regardless of the operational error created by manual digitizing, it is still the most common method of digitization used today and probably will remain so for the next several years.

SOURCES OF MAP ERROR

Sources of error in automated cartography are broken down into two primary components, source map error and operational error. Error is first introduced into the system by the error in source maps and data. Operational error can be further subdivided into error produced by data input and error produced by data manipulation (Walsh et al., 1987). Source map error includes the age of the maps or data, the extent of the areal coverage, the appropriate map scale, the map line width, and shrinkage and expansion of the paper map. In addition, the positional accuracy of the source map and the attribute or feature accuracy are major considerations in source material error. Positional errors are caused by errors in fieldwork or ground control and by error in the conventional map making process. All of this source map error is present prior to any conversion to a digital form where operational error is introduced (Burrough, 1986).

Conversion of the source map to a digital map during data input is where the largest operational errors are hypothesized to occur. During data input, errors arise from problems including map registration, digitizing, and rasterization. As a part of the data input errors, digitizing error is assumed to be significant, but it has not been well studied. Quite often the resolution of digitizing tables is quoted as a statement of accuracy; however, resolution is the smallest measurement produced and not the accuracy. Accuracy is defined as the expected error of the measurements in the long run (Chrisman, 1982). Many digitizing tables have re-

solutions of .001 inches, but the accuracy of the resulting digital maps are not necessarily to within .001 inches.

Digitizing error can be divided into line following error and line sampling error. Line following error is simply the operator inability to perfectly trace the map line with cursor. Line following error is caused in part by problems in locating true boundaries since drawn boundaries have some given line width and are not indefinitely thin. The actual boundary should be the centerline of the line; however, it is realistic to state that the map area occupied by a boundary line is rather ambiguous. Psychological and physiological factors effect line following error. Psychological errors are caused by the operator's perception of the map line he/she is following and the cursor crosshairs. Physiological errors stem from the inability of human muscles to keep the cursor on the map line (Jenks, 1981). The second part of digitizing error, line sampling errors, are errors dealing with the selection points to be used to represent the map line. This is also known as generalization error. The number and locations of points sampled and any line generalization algorithms are all sources of line sampling error.

The other type of operational error is data manipulation error. Processing, or data manipulation, errors include error caused by map overlays, vector to raster conversions, and classification and generalization. Most topological analyses have wrongly assumed that source data is uniform, digitizing is infallible, map overlay is merely intersecting boundaries and reconnecting a line network, and boundaries can be sharply defined and drawn. Rounding error and numerical precision are also sources of processing error (Burrough, 1986).

MODELS OF MAP ERROR

Map Error in General

Several models have been developed to model map line error and to model the cartographic line. These models provide background knowledge for studying line following error and digitizing error. Perkal (1956) developed an epsilon band width model to describe the error about a map line. He postulated that there existed a band of constant width around the map line. Points composing the line could theoretically occur anywhere within the epsilon band and still be legitimate points for defining the line. Blakemore (1984) tested Perkal's model further by using the epsilon banded line to bound a polygon. Points were randomly selected and tested to determine if they were in the polygon. Based upon the epsilon band theory, the selected point could be definitely in the polygon, possibly in the polygon, ambiguous if it fell near the center of the band, possibly out, and definitely out (Figure 1). With a epsilon band width of .7 km, only 50% of the tested points were definitely within the polygon.

Weaknesses of Perkal's (1956) model have been described by several authors. The model assumed every point of a line was within the band, but actually some points may lie outside the band. The epsilon band model also did not provide for a distribution of error within the band (Goodchild and Dubuc, 1987). Goodchild and Dubuc (1987) also observed that Perkal's model was not a good model of a line, but rather it only modeled the deviation for a point on the line. They stated that an error model should deal with a line as a continuum with strong autocorrelation. Peucker (1975) expanded the band width model by defining a line as having

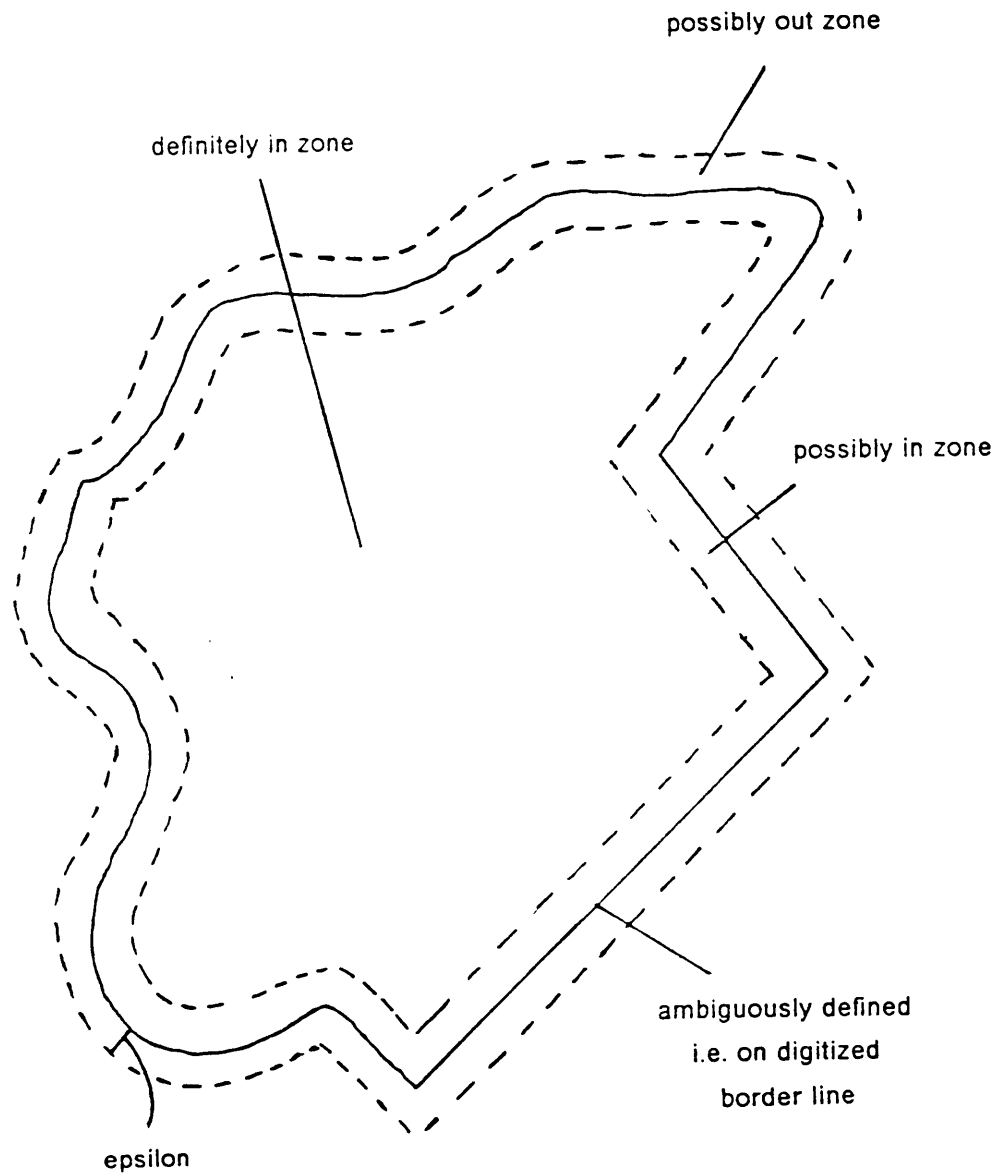


Figure 1: Blakemore's epsilon bounded point in polygon check.

1. a general direction
2. a band width, and
3. a length.

The band is represented by a bounding rectangle (Figure 2). At the highest level of abstraction, the line is represented by one rectangle. At the lowest level it can be represented by n rectangles, n being the number of points on the line. In this model, the bands for different line segments can vary in width (Peucker, 1975).

The various components of map error, source map error, registration error, digitizing error, and processing error, are additive. Each source is independent and the variances of the error distributions can be added together. Chrisman (1982) combined the variances of line width error, digitizing error, rounding error, and other error to create an "epsilon band". His model describes a locus of points around a line which represent the variance. Chrisman (1982) found that when tested, approximately 7% of the total polygon area fell within his epsilon band.

The geometry of a line has been represented as a plot of the orientation angle of line (ψ) against the cumulative curvilinear distance along the line (s) (O'Neill and Mark, 1987). Perkal's (1956) model has also been used to distinguish between spurious and real sliver polygons (Honeycutt, 1986). This study also found evidence that digitizing tends to produce a bimodal distribution of error. Models have also been developed for choropleth map error in general. Goodchild and Odette (1987) simulated choropleth images using spatially autocorrelated continuous variables and continuous variables generated from fractional Brownian surfaces. MacDougal (1975) proposed an estimate of total boundary inaccuracy as

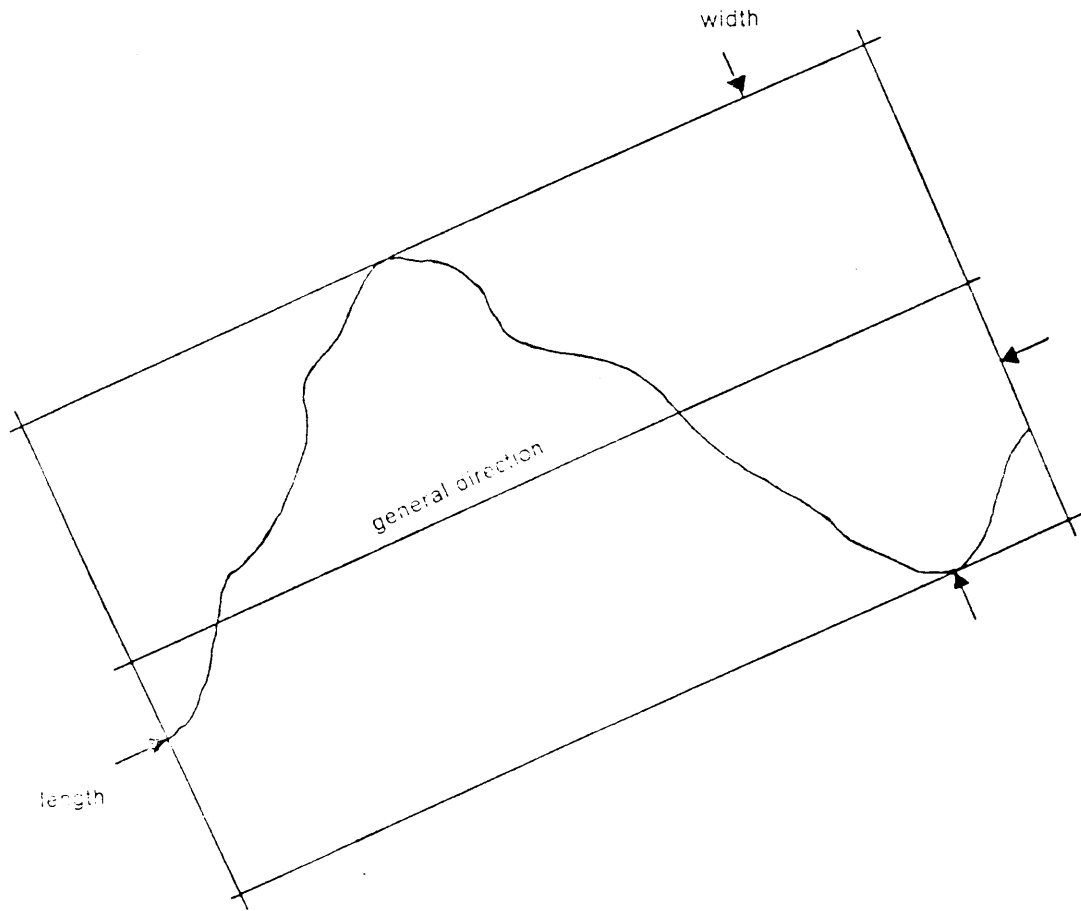


Figure 2: Peucker's theory of the cartographic line.

$$H = \sum_{i=1}^N \frac{h_i l_i}{T}$$

where h_i is the horizontal error (in standard deviations) of a line i , l_i is the length of line i , N is the number of boundary lines on a map, and T is the total map area.

In recent years, fractal geometry has been used to explain boundaries. From the study of fractals, it is clear that boundaries are not absolute, but rather they are compromises and are functions of scale, mapping method, and variance of property being mapped. Increasing map scale only reveals variation that had passed unnoticed before (Mandelbrot, 1982; Burrough, 1986; Goodchild, 1980).

Digitizing Error

Only a few studies have been done to quantify the effect human line-following error has on the accuracy of a digitized map. Many studies which use hand digitized maps simply dismiss the digitizing error. It is assumed human digitizing error will compensate for error inherent in the source materials and that the two are not cumulative (Boyle, 1970). Cartographic studies by Jenks (1981), Baugh and Boreham (1976), Ottawa (1987), and Traylor (1979) have attempted to examine the magnitude and processes of line following error. Several studies from the field of psychology have also looked at line following error, or tracking error.

Psychologists have studied the ability of a person to trace a line accurately although not in a cartographic setting. Studies have examined the ability for a person to track lines with a pen (Poulton, 1962). Other psychologists have attempted to understand and improve the

operator's ability to track targets (Conklin, 1957). Poulton (1962) also observed an operator's tendency to undershoot and overshoot depending on whether line changes were anticipated or not anticipated. Psychologists have not agreed upon a single theory that explains the motor coordination system, but these studies are helpful in describing line following error in a mapping situation (Traylor, 1979).

Jenks (1981) suggests that digitizing errors are not random, but they are highly correlated to the direction of movement of the cursor. Individuals have a personal and repetitive error signature. He also attributes digitizing error in stream mode to psychological error and physiological errors. Stream mode digitization contains error in both the longitudinal direction and the latitudinal direction although latitudinal errors are most common. Conversely, Jenks found longitudinal errors to dominate in point mode digitizing.

Baugh and Boreham (1976) examined the error in digitizing the coastline of Scotland. They suggested that digitizing produces a random error for a given point which introduces systematic error into the calculations for the length of the line. After studying digitizing procedures, Baugh and Boreham estimated error of positioning for an individual point to be $.10 \pm .05$ mm with a digitizer resolution of 0.1 mm. Another similar study found the standard deviation of a single point to be $\pm .18$ mm with a digitizing table resolution of .03 mm (Petersohn and Vonderohe, 1982).

In a study by Ottawa (1987), several inexperienced digitizers digitized polygons on a soils map. For a given polygon, the acreage as digitized by a particular individual varied by $\pm 7\%$ of the group mean acreage. The 7% variation was consistent for all polygons tested regardless of the polygon size. Ottawa concluded that human factors do affect digitizing accuracy, and that digitized data may not reach the accuracy level required by a map user. He also cited the need for continued research in this area and particularly the development of a model to predict digitizing errors.

One of the more in depth studies of digitizing error was by Traylor (1979). Traylor's purpose was to develop a methodology for studying line following error by accurately measuring and modeling digitizing error. A further objective of this study was to develop a program for training digitizers to correct their error based on a error signature. Error was empirically measured as the perpendicular distance between the original source map line and the digitized line. The sign of the error was determined, and he recorded the direction the cursor should have been moving as each sample was taken.

Traylor (1979) found that digitizing error was non-random, and that there was a systematic pattern to line following error. He developed a non-linear regression model to predict the error pattern. From this model a training program was developed to help digitizers reduce the error magnitude. Traylor's study emphasized the causative factors of digitizing error, but did not directly address the effect digitizing error had on the accuracy of the map product.

SUMMARY

The review of literature reveals that research has not focused on evaluating the effect of digitizing error on map accuracy. Digitizing errors are often treated as random occurrences with relatively unknown effects. General map error has been studied, and error sources have been identified and defined. Various theories describing the cartographic line and its associated error band have been presented, but none adequately address line-following by human digitizers. Psychologists have studied tracking error, but not in a cartographic situation.

One study has examined the causative factors of human digitizing error and developed a model of digitizing error. Modeling digitizing error has not been studied adequately although several authors expressed the need for such studies. In addition, it is basically unknown as to what quantitative effect digitizing error has on map accuracy in user-applicable terms.

MATERIALS AND METHODOLOGY

The project was divided into three phases. Phase one consisted of collecting data on digitizing error and statistically modeling that error. Using the statistical model from phase one, a simulation program was developed in phase 2 that simulated digitizing error and measured its effect on map accuracy. In phase three, the simulation program was used to determine how various parameters affect digitizing error and subsequently map accuracy. Some of these parameters included digitizing accuracy standards, size of digitized features, shape of digitized features, and complexity of digitized features.

PHASE I: Measuring and Modeling

Data Collection

Five individuals digitized a series of features to obtain a sample of digitizing error. Each digitized version of the feature was compared to a carefully digitized version considered to be the standard. Digitizing error was measured as the distance from the sample feature to the perpendicular intersection with the standard feature.

The test features consisted of 16 hand drawn lines of varying length and complexity. The features ranged in length from three inches to seven inches with approximately one - half inch difference in length between the lines. For every particular line length, there were two lines, with one line being more complex, or curvy, than the other. Whether or not a line was complex was a subjective determination. Table 1 shows a list of the test features and their actual length.

Table 1: Length of test features for digitizing.

Less Complex		More Complex	
ID Number	Length (in)	ID Number	Length (in)
1	3.16	9	3.34
2	3.69	10	3.57
3	4.13	11	4.21
4	4.74	12	4.76
5	5.09	13	5.07
6	5.37	14	5.46
7	6.02	15	6.08
8	7.08	16	7.06

A copy of the digitized features is shown in Appendix A. All the lines had a minimum length such that at least 50 points were recorded when the line was digitized by the operators in stream mode. The standard version of the features was digitized in point mode with the distance between digitized points being kept as small as possible, approximately 1mm. This standard was considered the true digital representation of the line for the purpose of measuring digitizing error.

The five digitizing operators had some prior digitizing experience, but they were not expert or "professional" digitizers. Each operator digitized the entire set of test features once using stream mode and once using point mode. In point mode, the operators moved the cursor along the line and selected when to digitize a point based upon their judgment. The operators were instructed to digitize the lines as if it were a production situation.

The digitizing table used was a Calcomp 9100 with a resolution of .001 inches and a repeatable accuracy of .005 inches. The table was linked to a Data General Desktop Generation Model 30 desktop computer. The digitizing software was the data input module of the geographic information system COMARC. The COMARC data files were converted to ASCII files containing the feature coordinates and downloaded to an IBM PC-AT for further analysis.

Measurement of Digitizing Error

Two aspects of digitizing error were measured. First, the perpendicular distance from a point on the sample line to where the perpendicular intersected the standard line was calculated. Secondly, the error was assigned a sign of positive or negative depending on which side of the standard the sample line was. A computer program written in MS-FORTRAN 77 calculated these values (Appendix B1). Many of the algorithms used to measure the error were based upon work done by Traylor (1979). Figure 3 provides a visual description of the perpendicular error distance. This error was determined mathematically by first calculating the location of the intersection of the perpendicular with the standard line segment by determinant solution of the two line equations. This location was calculated in three steps.

Step one of locating the intersection involved defining an equation for the perpendicular error distance. The standard line segment is vector A, and the perpendicular error distance is vector B (Figure 4).

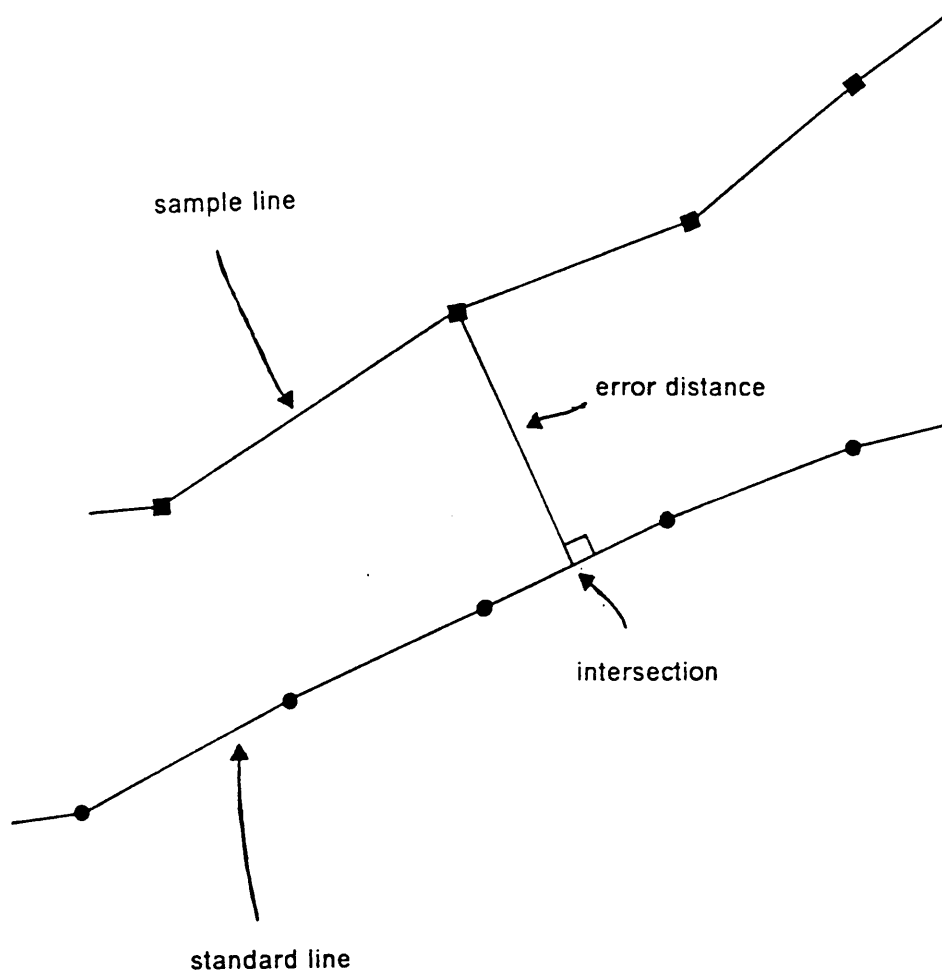


Figure 3: Perpendicular error distance between standard line and sample line.

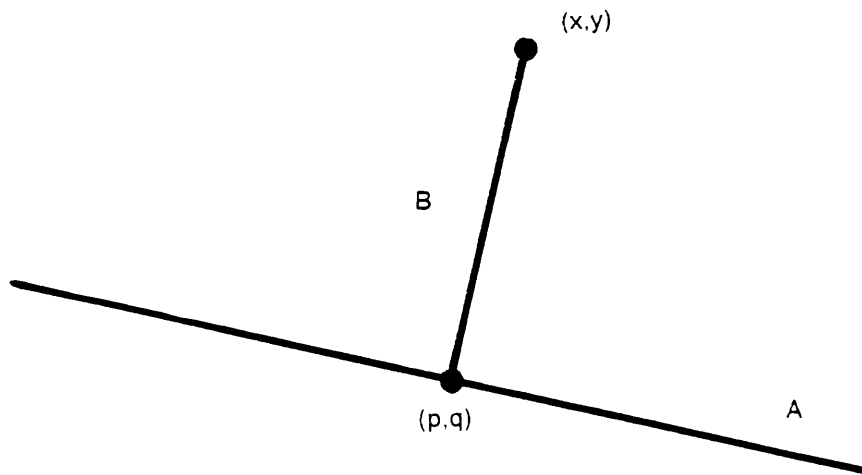


Figure 4: Intersection of vector A (standard line segment) and vector B (perpendicular error distance).

In order for the two vectors to be perpendicular, their product must be zero. The following equation was solved to define an equation for the perpendicular error distance:

$$A \times B = 0$$

$$[a,b] \times [x - p, y - q] = 0$$

In step two, the two-point-form equation

$$x - y_1 = (y_2 - y_1)/(x_2 - x_1) \times (x - x_1)$$

was used to define an equation for the standard line segment. Step three involved solving the two equations, the perpendicular error distance equation and the standard line segment, simultaneously to obtain their intersection. The X,Y coordinates of the intersection were then used to calculate the distance between the intersection and the point on the sample line.

The computer program simultaneously accessed the standard file and the sample file. The program selected the line segment formed by the first two points of the standard file. A line from the first sample point was dropped perpendicular to the first standard line segment, and the intersection location was determined. Several possible locations of the intersection were possible (Figure 5). In case 1 and case 2, the intersection fell within the standard line segment and was a valid intersection for the current line segment. The error distance was then calculated and the next sample point intersection was determined. Case 3 indicated that the sample point was ahead of the current line segment, and the program moved to the next standard line segment and recalculated the intersection for the line from the sample point. Case 4 occurred in two situations: 1) if the subject began digitizing before the beginning of the test feature, and 2) if the sample point occurred in a position such that a perpendicular could not be constructed to intersect a standard line segment (Figure 6). Case 4 was a rare occurrence, and any points falling into this category were dropped from consideration. This procedure was continued until the end of the sample line was reached. The sign of the error was also determined. Positive errors were defined as errors to the left of the line segment when viewed in the direction of digitizing. Those errors to the right of the line segment were considered negative errors.

Statistical Model

Inasmuch as stream mode produces a series of points that are close together in space and time, it was hypothesized that the stream mode data are serially correlated, or autocorrelated. This means that the magnitude and direction of the error of a digitized point are influenced by the magnitude and direction of the preceding points. As an operator continuously moves the cursor along the map line, he or she never perfectly follows the line, but instead continually crosses from one side of the line to the other side. As the operator strays

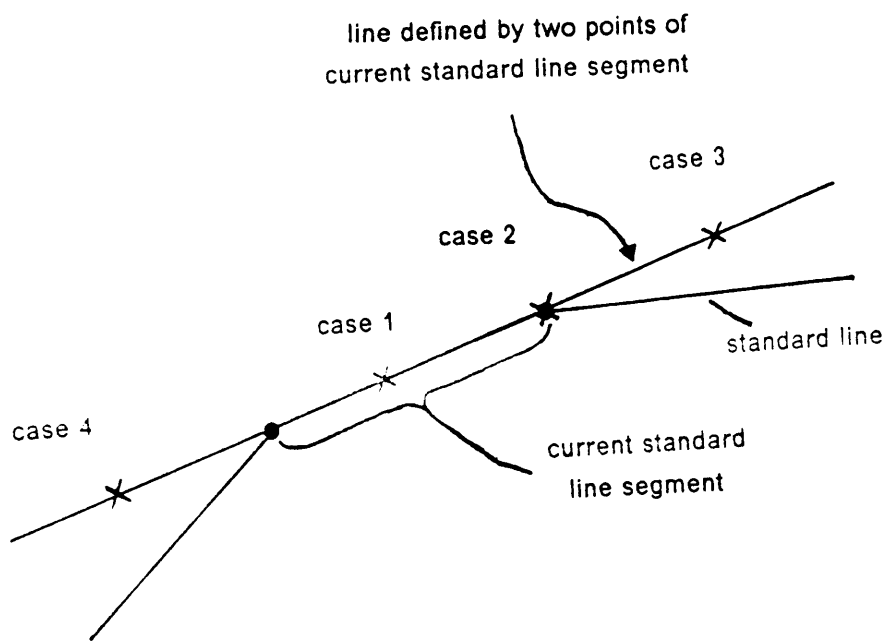


Figure 5: Possible locations of perpendicular intersections.

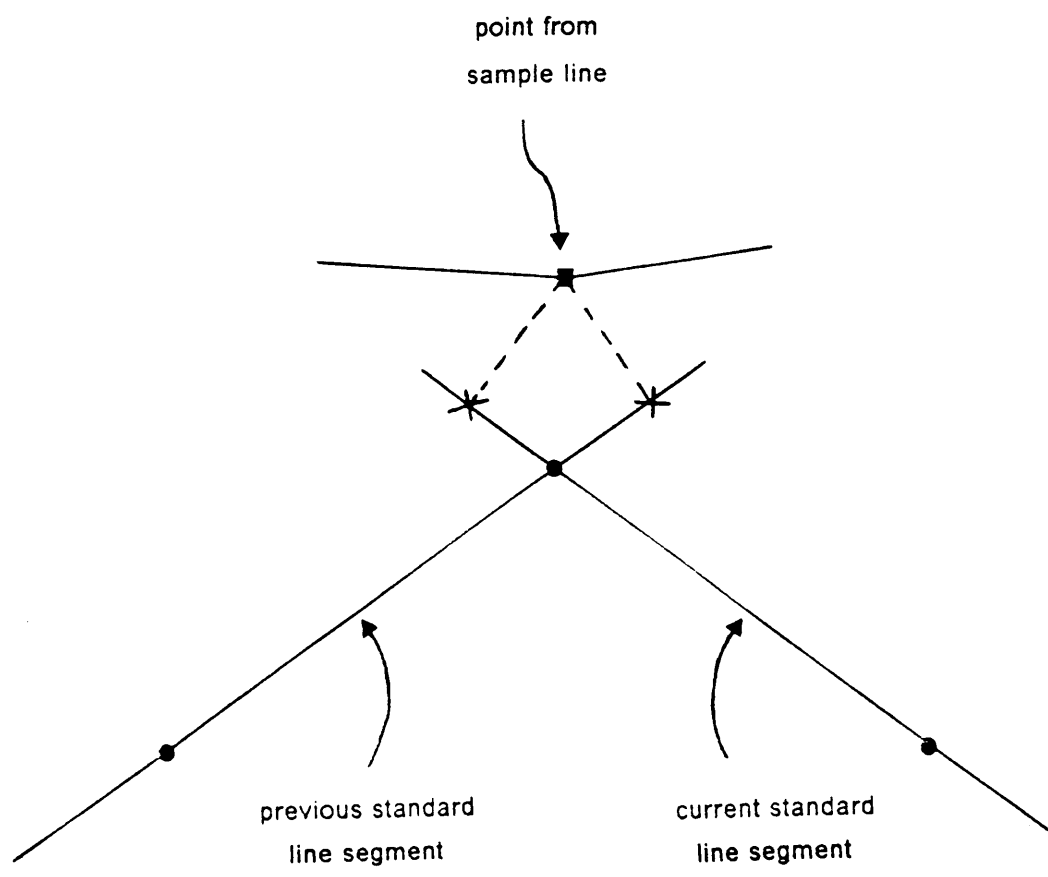


Figure 6: Perpendicular intersection case 4 where perpendiculars can not be constructed

from the line and recognizes it, a correction is attempted to return to the line. In contrast, the error in point mode digitizing was hypothesized to be uniformly random and not serially correlated. If the operator pauses at each point and aligns the cursor with line, the error at that point is not correlated with any previous point's error. With point mode, each point is an independent sample of the true position of the line. The operator is not continuously moving the cursor and constantly recording points; therefore, the operator does not attempt to compensate for error as the cursor is moving.

It was hypothesized that stream mode digitizing error can be statistically modelled as a time series. A time series is defined as a set of observations generated in time. The observations are usually dependent, such that the value of a particular observation is affected by previous observations in the series as well as a purely random component called white noise (Box and Jenkins, 1976). Stream mode digitizing can be modeled using time series analysis because of two assumptions 1) points are recorded at regular time intervals or regular distance intervals, and 2) the data are not random, but instead exhibit serial correlation. The concept of stationarity is important in time series modeling. A stationary time series is in equilibrium about a constant mean. The assumption of a constant mean for stream mode digitizing errors is justified because errors of a trained operator should occur, on average, on the left side of the line as frequently as they occur on the right side of the line and be similar in magnitude. Therefore, the mean of the series will be constant, namely zero. Nonstationarity implies, among other things, that an operator tended to favor a particular side of the line, a trait that presumably would be corrected by training. For this study, trained, experienced digitizers were used who did not have a digitizing bias, thereby removing operator differences and allowing use of stationary time series analysis.

The stochastic time domain approach using autoregressive integrated moving average (ARIMA) models was used to model digitizing error. A pure autoregressive (AR) process is one where the current point's error is expressed as a finite, weighted sum of p previous errors

plus a random shock a . A pure moving average (MA) process describes the current point's error as a finite number q of previous shocks (a 's) (Box and Jenkins, 1976). A moving average process is plausible since the operator can recognize errors and correct for them as he is digitizing. It was hypothesized that digitizing error can be described as a pure AR, pure MA, or a combination of autoregressive and moving average (ARMA). The general ARIMA(p,d,q) model reduces to simply an ARMA(p,q) model because the parameter d is zero due to stationarity of the digitizing error time series. The parameters, p and q , describe the order of the ARMA process. The order of the autoregressive process is represented by p , and q represents the order of the moving average process.

Literature was not available on using time series analysis to model digitizing error. Many other phenomena, particularly natural phenomena, have been described using time series analysis. Monserud (1986) used a Box-Jenkins ARMA model to analyze tree ring chronologies. Using time series analysis to model digitizing error assumed that digitizing error is not a random occurrence, but that it is correlated, meaning previous errors have an effect on the current error. Therefore, prior to fitting an ARMA model, the pattern of digitizing error was tested for nonrandomness. Separate tests for nonrandomness were done on the data digitized both in stream mode and point mode. Data from all five operators were combined for these tests. A standard runs test and a Chi-square test were used as tests for non-randomness (Zar, 1984).

Test for Non-randomness

The null hypothesis was that digitizing errors are random in nature. The runs test required: the number of points to the left of the standard line (NL), the number of points to the right of the standard (NR), and the number of runs (R). The expected number of runs was calculated as:

$$ER = [2(NL \times NR)/(NL + NR)] + 1$$

the variance was determined as:

$$VR = (ER - 1) \times \frac{[2(NL \times NR) - (NL + NR)]}{[(NL + NR) \times ((NL + NR) - 1)]}$$

The standardized z score was

$$Z = (R - ER) / \sqrt{VR}$$

If the runs were random, the z score fell within 1.96 at the 95% confidence level. Consistent z scores outside of that range caused rejection of the null hypothesis and indicated that digitizing error was nonrandom. If the error data were nonrandom, fitting of the ARMA model was pursued. If the error data were random, as was hypothesized for point mode digitizing data, a model was constructed without the use of an ARMA process. A random error indicated that the error of a point was not correlated to any previous error.

Another means of testing for the non-randomness of the error data was to observe the Chi-square values as produced by the identification stage of the ARIMA procedure in SAS (SAS Institute Inc., 1982). The null hypothesis was that the data are random. Large Chi-square values caused rejection of the hypothesis and indicated non-random behavior of the data.

Model Specification and Fitting

Using the identify step of the ARIMA procedure in SAS, the model parameters for a stream mode model were determined. The ARMA(p,q) model's parameters of p and q were

identified using techniques described by Box and Jenkins (1976). This was accomplished by examining plots of autocorrelations and partial autocorrelations. Based upon visual inspection and evaluation of these plots, a set of parameters was selected for further testing. This selection process is difficult to describe as it is a rather subjective evaluation of the plots of the correlation values. Once potential parameters were selected, estimates for the parameters were calculated and plots of the residual autocorrelations and residual partial autocorrelations were examined to determine if the selected parameters were correct. If the model was able to explain the correlation in the data well enough such that the magnitudes of the residuals were less than two standard deviations from zero, the model was assumed to be correct. Once an ARMA model was selected with the proper parameters, it was fit to all the data sets such that coefficients were derived for each line segment for each operator for a total of 80 coefficients

PHASE II: Simulation

Overview of Simulation

A simulation program was written in MS-FORTRAN 77 which simulated digitizing error by producing an error influenced version of a standard digitized feature (Appendix B2). The program simulated lines and polygons at various digitizing accuracy standards. The simulated lines and polygons were based upon a standard line or polygon which was digitized and assumed to be correct. For polygons, the area of the simulated polygon was determined and compared to the true area of the standard. The area was calculated using a standard algorithm for calculating the area of irregular shapes using Cartesian coordinates (Stolk and Ettershank, 1987). Run length distance was used as the measure of comparison for linear features. Run length distance was calculated as the sum of all the straight line lengths be-

tween each pair of coordinates. From observing the differences in areas and lengths over many simulations, a statement of digitized map accuracy was made in terms of area or length bias and the associated variance.

Accuracy standards for digitizing were represented by a stated band width at map scale and an alpha level. The alpha level was the maximum proportion of points allowed to fall outside the band width. This was a system of standards very similar to the National Map Accuracy Standards used by USGS (Thompson, 1979). For example, an accuracy standard of .01 at alpha level .05 means that 95% of the digitized points of a line or polygon must be within .01 map inches of their position on the source map.

The mechanics of simulating the digitizing error were very similar to reversing the process of measuring digitizing error. The simulation used the stated accuracy standard to set the bounds on the size of error to be created. Using the accuracy standards and the ARMA model, a set of errors was generated according to the scale of the feature being simulated. For every point on the standard feature, a corresponding simulated "error coordinate" was created by turning perpendicular to the standard line and moving the distance of the generated error.

Error Generation Process

For each simulation feature to be examined, a set of errors was generated. The error generation process of stream mode error was dependent upon the outcome of the ARMA fitting process. The following error generation process for a stream mode error simulation was an example of an ARMA(1,0) process. The form of the error equation was as follows:

$$E_i = \rho \times E_{(i-1)} + V_i \quad \text{eq.1}$$

where

E_i = serial correlated error of point i , $E_i \propto N(0, \sigma_e^2)$

ρ = correlation coefficient

V_i = random error of point i , $V_i \propto N(0, \sigma_v^2)$

$$\sigma_e^2 = \frac{\sigma_v^2}{(1 - \rho^2)} \quad \text{eq. 2}$$

This equation followed from the ARMA process which implied that the error of point i (E_i) was affected by the error of the previous point (E_{i-1}) plus some random error (V_i).

Step one in generating an error series was selecting a value for ρ (correlation coefficient). This value was determined based on the results of fitting the ARMA model in phase one. Eighty coefficients were produced from fitting 16 data sets for five operators. The empirical distribution of these 80 coefficients was assumed to represent the true distribution of the correlation coefficients. The empirical distribution was arbitrarily divided into six categories and a probability assigned to each category based on the frequency of correlation coefficients. The mean of the range of correlation coefficients for each category was assigned to that category as the correlation coefficient for that particular category so that there were six correlation coefficients and six categories. For each error series generation, one of the six correlation coefficients was selected by first selecting a random number between 0 and 1. The random number corresponded to one of the probability ranges assigned to each category of the empirical distribution. The correlation coefficient for that category was then used in the simulation for that error series.

Step two was to generate the variance of the random error (σ_e^2) and the variance of the error term (σ_e^2). Variance of the error term was calculated first and then entered into equation 2 to solve for the variance of the random error. Variance of the error was determined by calculating the standard deviation of the error (σ_e). The following equation was used to determine σ_e .

$$\sigma_e = \frac{W}{K} \quad \text{eq. 3}$$

where

$W = 1/2$ of the accuracy standard band

$K = 1 - \frac{\alpha}{2}$ quantile of standard normal distribution.

where

$\alpha =$ alpha level of accuracy standard

Equation 3 was derived using the following reasoning. If $E \sim N(0, \sigma^2)$ then $Z = \frac{E}{\sigma_e} \sim W(0, 1)$. By definition, $P[|z| \leq K] = 1 - \alpha$, and by substitution, $P\left[\frac{|E|}{\sigma_e} \leq K\right] = 1 - \alpha$. Therefore, $P[|E| \leq K\sigma_e] = 1 - \alpha$. Now, the digitized errors, E , need to obey $P[E \leq W] = 1 - \alpha$, so by equality, $W = \sigma_e K$. The accuracy standard (W) was in the units of the standard feature; thereby scaling the error to be generated. Variance of the random error was simply calculated by entering σ_e^2 into equation 2 and solving for σ_e^2 .

Step three was to generate the values for the error distances. The initial error in a series is unaffected by any previous error and was defined as

$$E_o = \frac{V_o}{\sqrt{1 - \rho^2}}$$

where

E_0 = initial error

V_0 = random error, $V_0 \propto N(0, \sigma_v^2)$

ρ = correlation coefficient

The remaining errors were generated using equation 1. A value for V_i was generated by selecting a random number from a normal distribution and multiplying by σ_v . During the simulation, random numbers were generated using a multiplicative congruential random number generator (Balci, 1985).

Point mode error was generated by first selecting a random direction around the point to be simulated. Next, an error distance was selected by drawing from a normal distribution of error distances with mean zero and a variance of σ_e^2 where

$$\sigma_e^2 = \left(\frac{W}{K} \right)^2$$

where

W = 1/2 of the accuracy standard band

$K = 1 - \frac{\alpha}{2}$ quantile of standard normal distribution.

where

α = alpha level of accuracy standard

The simulation program allowed the user to input the accuracy standard band width and alpha level. From this information, the program generated an array of error values to be used to simulate the point coordinates of a line or polygon to be simulated were stored in a computer file to be accessed by the simulation program.

The simulation program took the first point of the standard feature and the first error distance of the error array. Using standard surveying trigonometry, the coordinate values were calculated for a point 90 degrees from the line segment defined by the current standard coordinate and the next standard coordinate at the given error distance (McCormac, 1985). The error coordinate was 90 degrees to the left or to the right of the standard feature depending upon the sign of the error distance. This process continued for every point on the standard feature.

Once all error coordinates were calculated for the feature, line length or area was calculated. Polygon area and line length were calculated using the same algorithms described previously. The length or area was stored in an array, and the process was repeated for the next simulation by generating a new array of error distances. After all the specified simulations had been generated for a particular standard feature, the program calculated a statistical summary. The statistical summary included the true area or length measurement, the mean simulated area or length, the bias (true - mean simulated), variance, standard deviation, root mean squared error, the maximum error distance, and the average absolute error distance. These statistics were expressed both in absolute and relative terms. All of these statistics were calculated for each simulated feature over all the simulations for the particular feature.

PHASE III: Analysis

In phase three, various lines and polygons were simulated using several accuracy standards. Simulated features varied in size and complexity. The results of the simulations were analyzed for the affects of accuracy standards, mode of digitizing, and feature characteristics.

Data Collection

Twenty-five polygons and 25 lines were simulated. These features were hand drawn and covered a range of sizes and shapes. The polygons ranged in size from .5 square map inches to 25 square map inches, and the lines ranged from 3 map inches to 12 map inches. The features included curvy and nonlinear features as well as straighter, linear features. The features were digitized and then entered into the simulation program. Each feature was simulated once using the point mode model and once using the stream mode model. The parameters for the standards are listed in Table 2. The standards were selected on the basis of currently used accuracy standards. Bandwidths of .01 inches and .02 inches were reasonable bandwidths based on the author's experience and the NMAS which is .02 inch bandwidth at .10 alpha level. The bandwidths of .005 inches and .04 inches are at the extremes of reasonable accuracy. The standards were also selected such that the bandwidths and the alpha levels increased by a factor of two. Each feature was simulated 200 times. This provided 200 different error influenced versions of the particular feature. The number of simulations was selected by running the simulation program for increasing number of simulations and observing where the variance in polygon size or line length began to stabilize. One hundred simulations appeared to provide sufficient stability of variance, but this number was doubled to ensure stability.

Analysis

For analysis, the values recorded for simulated acreage and footage were based upon a map scale of 1:24000. Bias and standard deviation of polygon area and line length were used to measure the effects of digitizing error on map accuracy. Bias was calculated by subtracting the simulated polygon size or line length from the true feature size. These biases were then averaged for the 200 simulations. Standard deviation was simply the standard deviation of the

TABLE 2: Map accuracy standard parameters.

Band width (map inches)	Alpha level
.005	.025
.005	.05
.005	.10
.005	.20
.01	.025
.01	.05
.01	.10
.01	.20
.02	.025
.02	.05
.02	.10
.02	.20
.04	.025
.04	.05
.04	.10
.04	.20

200 simulations. Relative standard deviation is also known as the coefficient of variation. Bias is a measure of accuracy in that it indicates how far from the true feature size a digitizer should be on average. Standard deviation is a measure of the spread around the mean digitized feature, or precision. The results of the simulations were analyzed further by plotting bias and standard deviation over feature size for polygons, and then for lines. From this analysis, the effect of line length and polygon size on bias and standard deviation was examined. This effect may be important since map features may have different potential for error depending upon their size. For the same feature, comparisons were made between stream mode and point mode simulations. These comparisons were made by observing differences in average length or area, bias, and standard deviation. All of the preceding analyses were done using all the various accuracy standards to evaluate the effect of varying accuracy standards. In Appendix D, bias and standard deviation are referred to as absolute bias and absolute standard deviation. In Appendix D, the quantities relative bias and relative standard deviation were calculated by taking the bias and dividing by the true feature size and by taking the standard deviation and dividing by the true feature size.

RESULTS AND DISCUSSION

The results and discussion are presented in three phases just as the materials and methodology section was subdivided. Phase one consists of the results from collecting data from several digitizers and fitting an ARMA model. Phase two is a very short section on the development of the simulation program. Finally, the results of simulating digitizing error are presented in phase three.

PHASE I: Measuring and Modeling

Empirical Data Analysis

The means and standard deviations of the errors were calculated for each feature digitized by an operator after five digitizers digitized the 16 test features and the perpendicular error distance was measured for every point. These values are presented in Appendix C. Table 3 presents a summary of means and standard deviations of digitizing error for each operator with all 16 features combined for stream mode and point mode respectively.

Table 3: Means and standard deviations of digitizing error over all features for each operator.

Operator	Number of points	Mean (inches)	Std.Dev. (inches)
Stream Mode			
1	1687	-.001237	.028556
2	1811	.013153	.046325
3	1432	.023453	.053065
4	1421	.016441	.046896
5	1397	.024414	.051085
all operators	7748	.014557	.046487
Point Mode			
1	2025	.001639	.022623
2	590	.022356	.043624
3	673	.019404	.039944
4	719	.017312	.041253
5	562	.017653	.035685
all operators	4569	.011367	.034778

Previously, it was assumed that the mean error was zero which would indicate stationarity in the data. Except for operator one, the error means shown in Table 3 did not initially appear to be close to zero, but instead, a bias to the positive side of the line seemed to exist. However, all the means were within one standard deviation of zero such that there was no significant difference from zero. Since the error means were not significantly different from zero, the assumption of non-stationarity and unbiased digitizing was upheld. Differences between point mode and stream mode were also evident from this data. The means of the error data showed little difference between point mode and stream mode digitizing, but the standard deviations were always smaller in point mode (Table 3). In other words, little difference existed in accuracy of digitizing between stream mode and point mode, but point mode was more precise.

Histograms of the error data were plotted for each operator in point mode and in stream mode. Figure 7 shows the histogram distribution of the digitizing error data from all features and all operators combined in stream mode. Figure 8 shows the distribution from point mode digitizing. These distributions appeared to approximate a normal distribution. This approximation of a normal distribution supported the assumptions of normality needed to use the ARMA modeling procedure.

Tests for Non-randomness

Plotting:

If a process is serially correlated, as was hypothesized for stream mode digitizing, the observations are not independent, and depending on the nature of the correlation, some type of pattern may be visually evident. Therefore, an initial way to determine if a process is serially correlated is simply to observe a graph of the observed variable over time. Figure 9 and Figure 10 show digitizing error plotted over digitizing interval for stream mode and point mode

Empirical Error Distribution

Stream Mode Combined

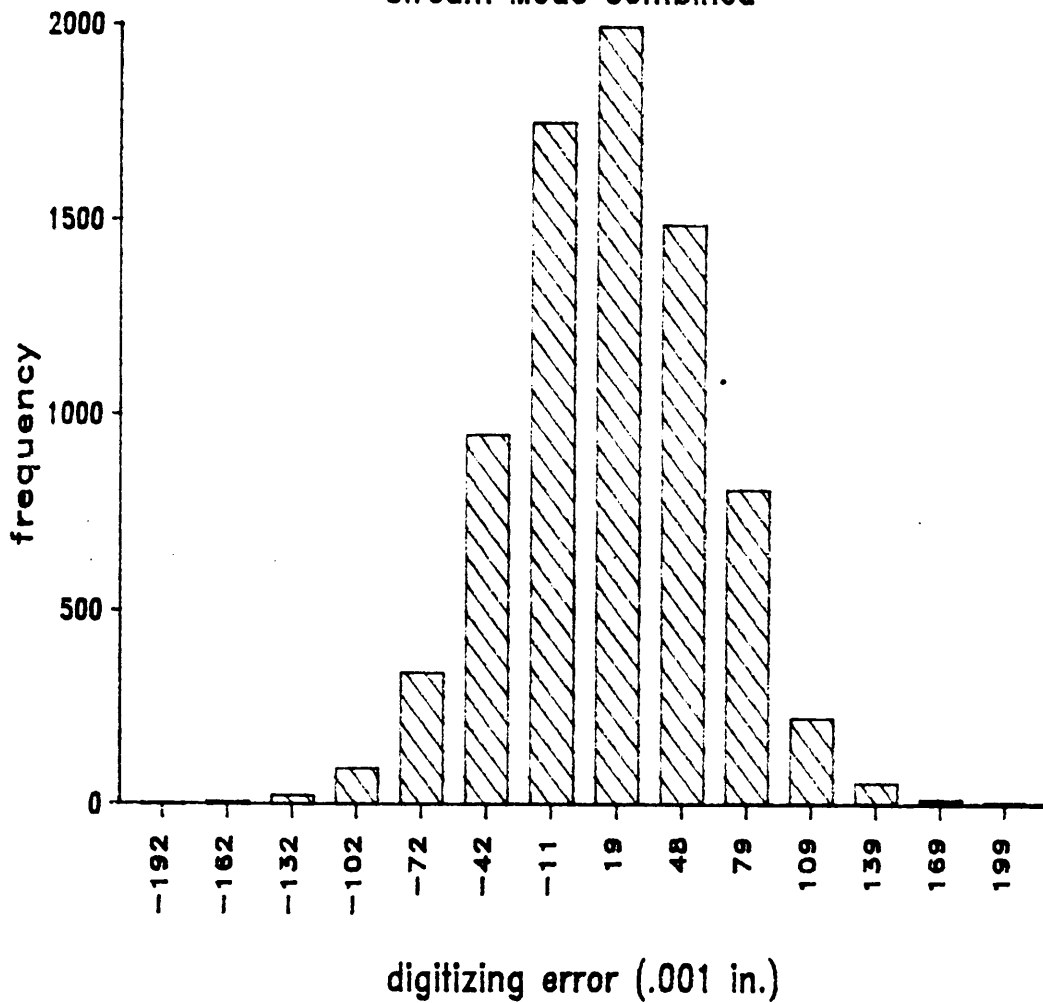


Figure 7: Distribution of stream mode digitizing error from five operators and 16 features; interval labels are uneven due to rounding error.

Empirical Error Distribution

Point Mode Combined

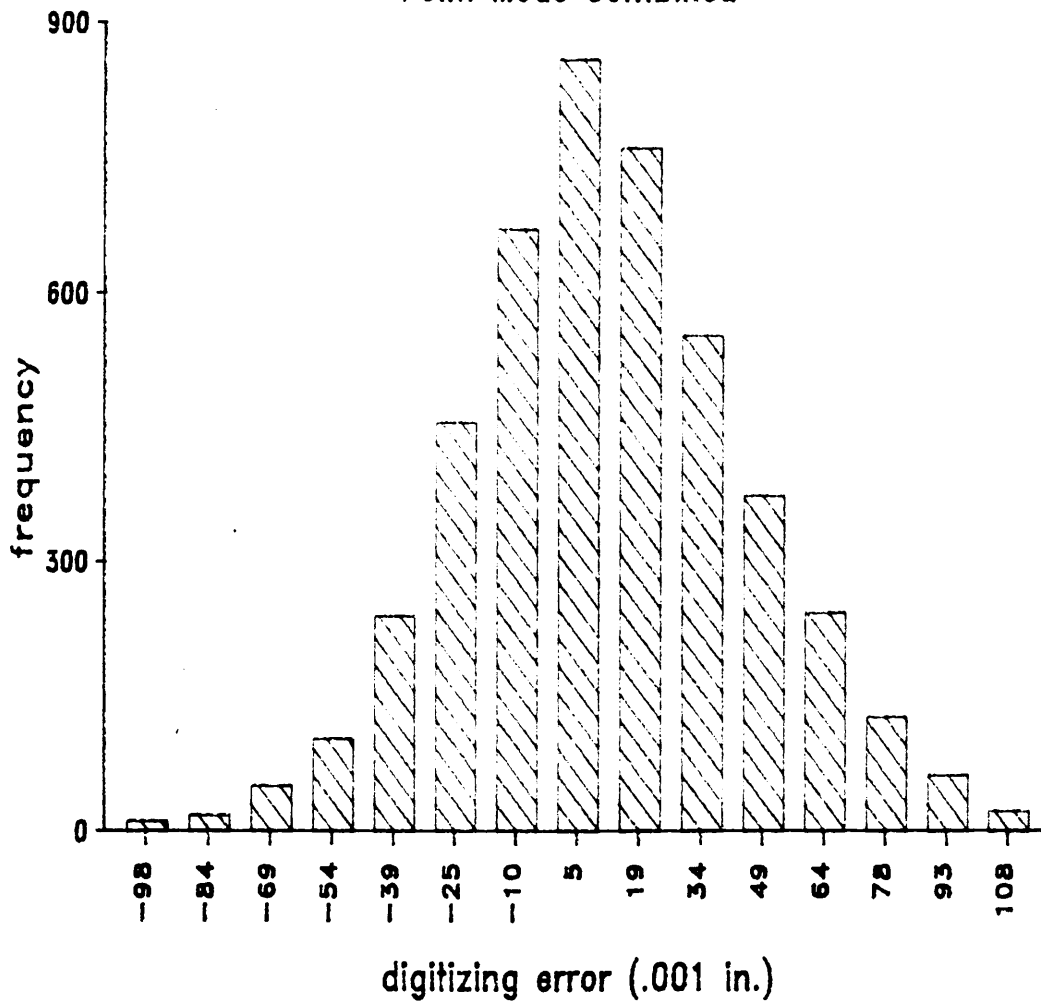


Figure 8: Distribution of point mode digitizing error from five operators and 16 features; interval labels are uneven due to rounding errors.

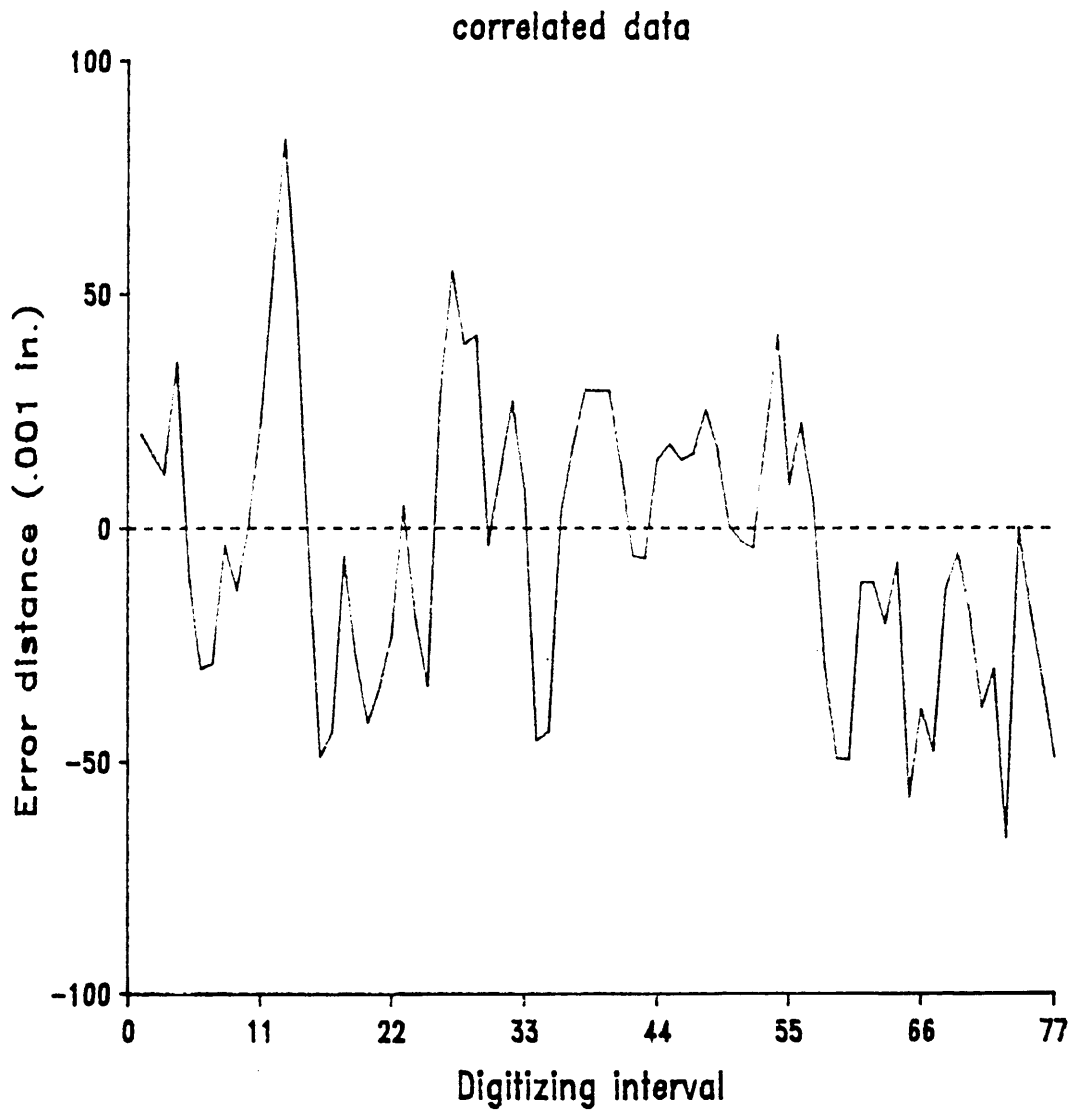


Figure 9: Plot of stream mode digitizing error over digitizing interval

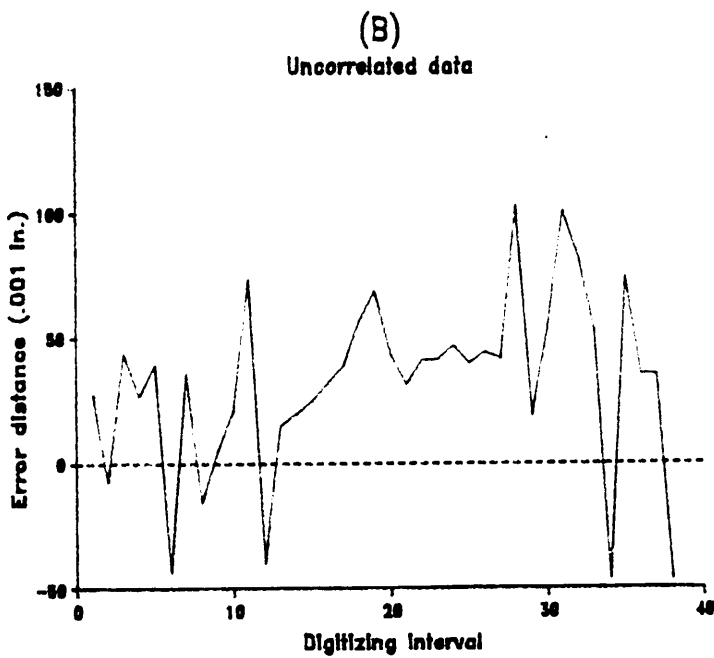
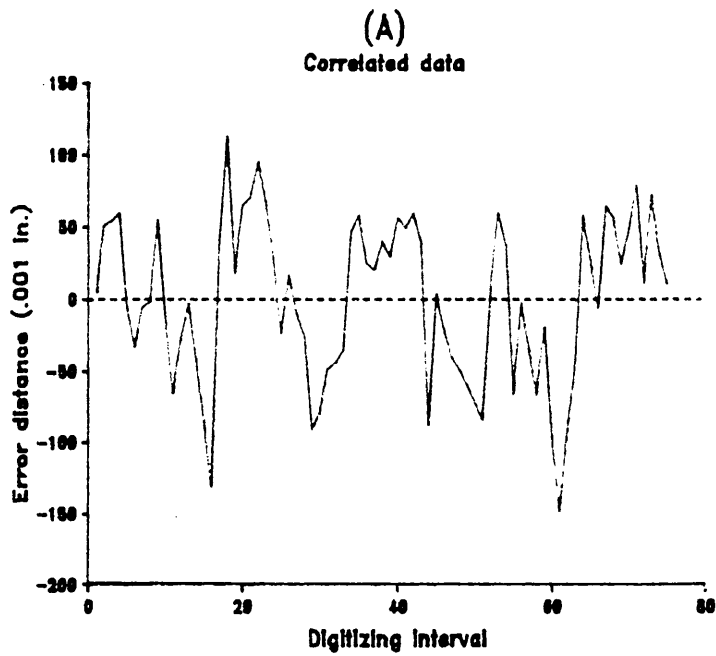


Figure 10: Plot of point mode digitizing error over digitizing interval

respectively. The non-random pattern exhibited by stream mode digitizing typifies an autocorrelated process for digitizing error (Figure 9). Figure 9 is the digitizing error for test feature 2 for operator 1. Figure 10a shows a plot of digitizing error from a line digitized in point mode (feature 16 of operator 4) which also exhibits serial correlation as the stream mode plot did in Figure 9. In contrast, Figure 10b shows a plot of digitizing error from a line digitized in point mode which does not exhibit strong serial correlation, but rather it shows the data to be more random. Figure 10b is a plot of the digitizing from test feature 11 of operator 4. The reasoning being that there was not a regular pattern to the "runs", or crossing of the zero line as compared to Figure 10a. Some point mode error data sets had patterns characteristic of serially correlated processes, and others had characteristics resembling random processes.

Runs test:

The z-scores for the results of the runs tests are shown in Table 4. The great majority of the stream mode tests rejected the null hypothesis since z-scores were greater than ± 1.96 (2 standard deviations) with the null hypothesis being that stream mode digitizing errors were random, or uncorrelated in nature. The z-scores for the runs tests on point mode error data are shown in the second part of Table 4. Two-thirds of the 80 tests rejected the null hypothesis at alpha of .05. These tests appeared to indicate that point mode digitizing errors can be either correlated or uncorrelated. In most cases the z-scores for both stream mode and point mode were negative values. Negative values indicated that the observed processes had fewer number of runs than a random process. This was expected due to the pattern produced by digitizing where several points are recorded for each run.

Chi-square test:

The final test to check for non-randomness was a Chi-square test provided by the identify step of the ARIMA procedure in SAS (SAS Institute Inc., 1982). In stream mode, the Chi-square values were always significant; therefore, the null hypothesis that the digitizing error data were uncorrelated was rejected. For point mode digitizing error, approximately half of the 80

Table 4: Z-scores for runs tests for stream mode and point mode digitizing.

feature number	operator 1	operator 2	operator 3	operator 4	operator 5
Stream Mode					
1	-5.237	-3.013	-6.233	-6.166	-5.206
2	-5.595	-5.894	-4.838	-5.678	-3.404
3	-4.451	-7.339	-3.908	-6.032	-4.400
4	-4.415	-9.922	-5.286	-6.526	-5.799
5	-4.634	-.882	-5.078	-3.658	0.000
6	-3.394	-7.771	-5.370	-6.808	-5.969
7	-1.857	-5.462	-2.863	-4.441	-3.007
8	-3.999	-9.578	-5.903	-7.155	-4.880
9	-3.043	-5.882	-4.444	-3.656	-2.452
10	-3.710	-5.454	-4.568	-1.789	-4.496
11	-3.357	-9.066	-5.308	-6.067	-4.916
12	-3.890	-9.528	-7.036	-6.876	-7.597
13	-3.493	-8.848	-7.110	-7.271	-7.583
14	-3.668	-6.620	-5.034	-6.352	-6.501
15	-6.301	-7.911	-7.975	-8.480	-6.220
16	-3.960	-7.129	-6.508	-6.563	-7.457
Point Mode					
1	-8.443	-1.301	-2.604	-2.218	-1.311
2	-4.675	-1.647	-2.826	-.758	-1.207
3	-4.989	-5.505	-2.197	-2.622	-1.564
4	-3.985	-5.668	-6.338	-3.164	-2.306
5	-4.214	0.000	-2.2020	-.589	0.000
6	-0.792	-6.644	-3.588	-3.361	-.132
7	-2.266	-1.022	-1.384	-1.629	.310
8	-2.815	-4.698	-2.938	-3.110	-4.783
9	-3.937	-4.232	-1.992	-3.346	-4.913
10	-4.520	-3.205	-.922	-.847	-3.072
11	-3.042	-5.918	-.067	-.918	-3.816
12	-4.429	-4.871	-3.505	-3.186	-4.507
13	-4.281	-5.402	-4.767	-3.548	-2.426
14	-4.645	-5.196	.343	-4.129	-.983
15	-1.940	-1.581	-1.768	-1.975	-2.062
16	-5.231	-5.918	-5.226	-1.944	-.222

features tested had non-significant Chi-square values which caused the null hypothesis to not be rejected at a confidence level of .05, and one half of the features caused rejection of the null hypothesis.

Discussion and summary:

The plots of digitizing error, runs tests, and Chi-square tests all strongly supported the hypothesis that stream mode digitizing was serially correlated; therefore, stream mode digitizing error could be modelled as a time series. However, the hypothesis that point mode error was always random was not supported by the results of digitizing error, runs tests, and Chi-square tests. Point mode error appeared to sometimes be correlated depending on the operator's subjective sampling of points and the length and complexity of the line. The hypothesized style of point mode digitizing, where each point is an independent observation, did occur part of the time; however, often in point mode digitizing, the error of one point is influenced by previous points, just as it is in stream mode. The degree of correlation in point mode digitizing seemed to be affected by the shape and complexity of the feature being digitized, and by the way in which point mode digitizing is defined by a particular digitizing operator. If an operator keeps the cursor moving and continuously presses a button to record a point, "point mode" then becomes a type of simulated stream mode, and consequently, the points become correlated.

In order to continue studying and modeling point mode digitizing error, only one of the types of point mode was selected for study. Point mode where each point is an independent, random observation uncorrelated with any previous points was the type of point mode selected. This was termed discrete point mode, as opposed to continuous point mode. For example, digitizing the four vertices of a square is an example of discrete point model. Point mode could then be modeled as originally hypothesized. However, the development of models to describe point mode digitizing where the errors are correlated deserves much further study.

Model Specification and Fitting

All 80 error data sets (16 features for five operators) were examined using the identify step of the ARIMA procedure in SAS (SAS Institute Inc., 1982). The majority of the data sets tested exhibited behavior characteristic of an AR(1) process. This was determined by the pattern produced by plots of the autocorrelations and partial autocorrelations using methods described by Box and Jenkins (1976). The data were then fit to an AR(1) model and plots of the residual autocorrelations and residual partial autocorrelations examined. The correlation present in the residuals after fitting the model was only white noise. Chi-square tests also showed the residuals to be random, or white noise. Therefore, most of the autocorrelation present in the digitizing error data sets was explained using an AR(1) model. SAS estimated a correlation coefficient, ρ , for each of the 80 data sets. The coefficients are shown in a histogram in Figure 11. The majority of the coefficients were in the .65-.85 range which indicated a strong correlation process in digitizing error. This empirical distribution was used later in the simulating process to select values for ρ .

No actual model fitting was needed for the point mode error model since point mode was to be modelled as a random, uncorrelated error. The point mode model consisted of modeling the error of an individual point as a random direction about the correct point, and an error distance drawn from a normal distribution with a variance defined by a particular accuracy standard.

PHASE II: Simulation

Phase 2 simply resulted in the development of the simulation program, SIMDIGER, which is shown in Appendix B2. The mechanics of the simulation are explained and outlined in the phase 2 section of the Materials and Methodology. The error generation procedure in the

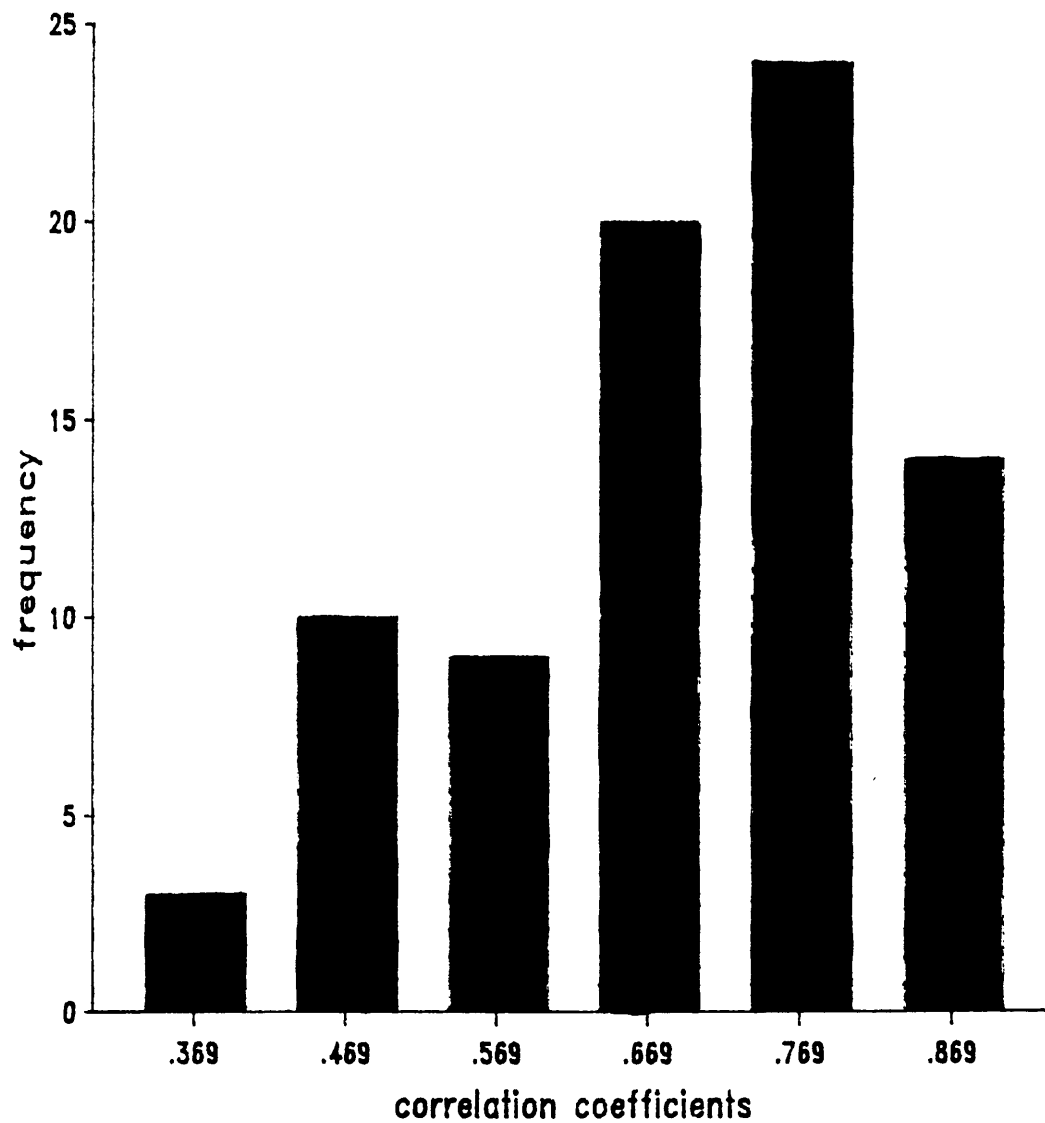


Figure 11: Distribution of correlation coefficients for stream mode digitizing error after fitting an AR(1) model.

Materials and Methodology section was used to simulate stream mode error using an AR(1) process. Values for the correlation coefficients were selected using probability list sampling from the distribution of correlation coefficients shown in Figure 11 as explained in the Materials and Methodology section. Point mode error was simulated as described in the Material and Methodology section using a random model approach.

PHASE III: Analysis of Simulation

Error Magnitudes and Trends Across Accuracy Standards

Bias and standard deviation were calculated for each feature using 200 simulations for each of the 16 sets of accuracy standards. These biases and standard deviations were then averaged across all 25 features simulated at each of the accuracy standards. This produced a mean bias and a mean standard deviation for each of the 16 accuracy standards, and these means are presented in tables 5 through 12.

Table 5 shows the average bias of polygon area for stream mode simulation. The bias was essentially zero for all alpha levels and accuracy standards. No trend in bias was apparent as accuracy standard bandwidth or alpha levels were changed. The results from simulating in point mode were very similar to stream mode in that the bias was very close to zero (Table 6). Evidently, there was a compensating effect on area estimates when digitizing polygons. That is, the increase in area caused by errors on one side of the line was approximately equal in magnitude to the decrease in area caused by errors on the other side of the line. Regardless of the digitizing accuracy standard, there was no difference between digitized polygon area and the source map polygon area in the long run.

Table 5: Average bias of polygon area for stream mode simulations a) bias in actual acres, and b) bias in percent of actual acres.

Average Bias of Polygon Area (acres)

		Stream Mode			
		Alpha level			
		0.025	0.05	0.10	.20
Standard (map inches)	0.005	0.003	0.004	0.001	0.015
	0.010	0.025	-0.021	-0.039	.017
	0.020	0.024	0.030	-0.099	-0.097
	0.040	-0.094	0.078	-0.038	-0.256

Average Bias of Polygon Area (% of actual acres)

		Stream Mode			
		Alpha level			
		0.025	0.05	0.10	0.20
Standard (map inches)	0.005	0.005	0.000	-0.001	0.006
	0.010	0.003	-0.003	-0.013	-0.012
	0.020	-0.007	0.013	-0.017	-0.052
	0.040	-0.038	-0.004	0.054	-0.080

Table 6: Average bias of polygon area for point mode simulations a) bias in actual acres, and b) bias as percent of actual acres.

Average Bias of Polygon Area (acres)

Point Mode

Alpha level

		0.025	0.05	0.10	.20
Standard (map inches)	0.005	0.007	-0.007	0.001	0.011
	0.010	-0.014	0.000	0.009	-0.001
	0.020	0.030	-0.001	-0.003	0.067
	0.040	-0.005	0.028	-0.026	0.036

Average Bias of Polygon Area (% of actual acres)

Point Mode

Alpha level

		0.025	0.05	0.10	0.20
Standard (map inches)	0.005	0.003	-0.001	0.001	0.004
	0.010	-0.005	-0.002	0.001	-0.008
	0.020	0.002	0.001	-0.002	0.007
	0.040	-0.002	0.000	-0.030	0.001

The average standard deviations for each accuracy standard bandwidth and alpha level are presented in Table 7 for stream mode, and in Table 8 for point mode. The standard deviations were not very large with the stream mode values ranging from .118 percent of actual area to 1.677 percent of actual area (Table 7b). The stream mode values were approximately 1.5 times larger than the point mode values which ranged from .075 percent of actual area to 1.044 percent of actual area (Table 8b). Figure 12 shows a comparison between the magnitudes of the standard deviations for stream mode and the standard deviations for point mode. Clearly, changing the digitizing accuracy standards affected the amount of variation, or precision, in area estimates, and bandwidth significantly had more of an impact than alpha level. The estimate of area for a particular polygon was more variable under less stringent standards. Doubling the accuracy standard bandwidth caused an almost exact doubling of the standard deviation. This was true for both the stream mode simulations and the point mode simulations. As the alpha level doubled, the standard deviation increased at the rate of 1.15 times to 1.3 times. Therefore, although digitizing at a less stringent accuracy standard resulted in no bias over many simulations, an individual simulation may have had some positive or negative error associated with it due to the lower precision, i.e. large standard deviation, at the less stringent accuracy standards.

Unlike polygon area error, line length error did not have a bias of zero. Table 9 shows the average bias for lines when simulated in stream mode. The biases changed dramatically as the accuracy standards were varied. The most stringent accuracy standard produced a bias of -.066 percent of actual line length while the loosest standard had a bias of -11.119 percent of actual line length. Point mode produced a similar range of biases except they were one-third as large in magnitude as the stream mode biases (Table 10). Figure 13 allows a graphic comparison between point mode and stream mode. Doubling the accuracy standard bandwidth caused a quadrupling of the bias for both point mode and stream mode. Doubling the alpha level only produced an approximate increase of 1.5 in the bias.

Table 7: Average standard deviation of polygon area for stream mode simulations a) standard deviation in actual acres, and b) standard deviation as percent of actual acres.

Average Std. Dev. of Polygon Area (acres)

Stream Mode

Alpha level

		0.025	0.05	0.10	.20
Standard (map inches)	0.005	0.470	0.543	0.638	0.813
	0.010	0.940	1.067	1.283	1.658
	0.020	1.891	2.127	2.541	3.321
	0.040	3.746	4.232	5.116	6.564

Average Std. Dev. of Polygon Area (% of actual acres)

Stream Mode

Alpha level

		0.025	0.05	0.10	0.20
Standard (map inches)	0.005	0.118	0.137	0.162	0.207
	0.010	0.235	0.269	0.324	0.418
	0.020	0.492	0.533	0.639	0.832
	0.040	0.938	1.077	1.284	1.677

Table 8: Average standard deviation of polygon area for point mode simulations a) standard deviation in actual acres, and b) standard deviation as percent of actual acres.

Average Std. Dev. of Polygon Area (acres)

Point Mode

Alpha level

		0.025	0.05	0.10	.20
Standard (map inches)	0.005	0.327	0.371	0.441	0.565
	0.010	0.651	0.746	0.901	1.130
	0.020	1.288	1.499	1.752	2.290
	0.040	2.612	3.003	3.618	4.561

Average Std. Dev. of Polygon Area (% of actual acres)

Point Mode

Alpha level

		0.025	0.05	0.10	0.20
Standard (map inches)	0.005	0.075	0.084	0.100	0.131
	0.010	0.148	0.172	0.206	0.257
	0.020	0.295	0.338	0.394	0.530
	0.040	0.602	0.702	0.811	1.044

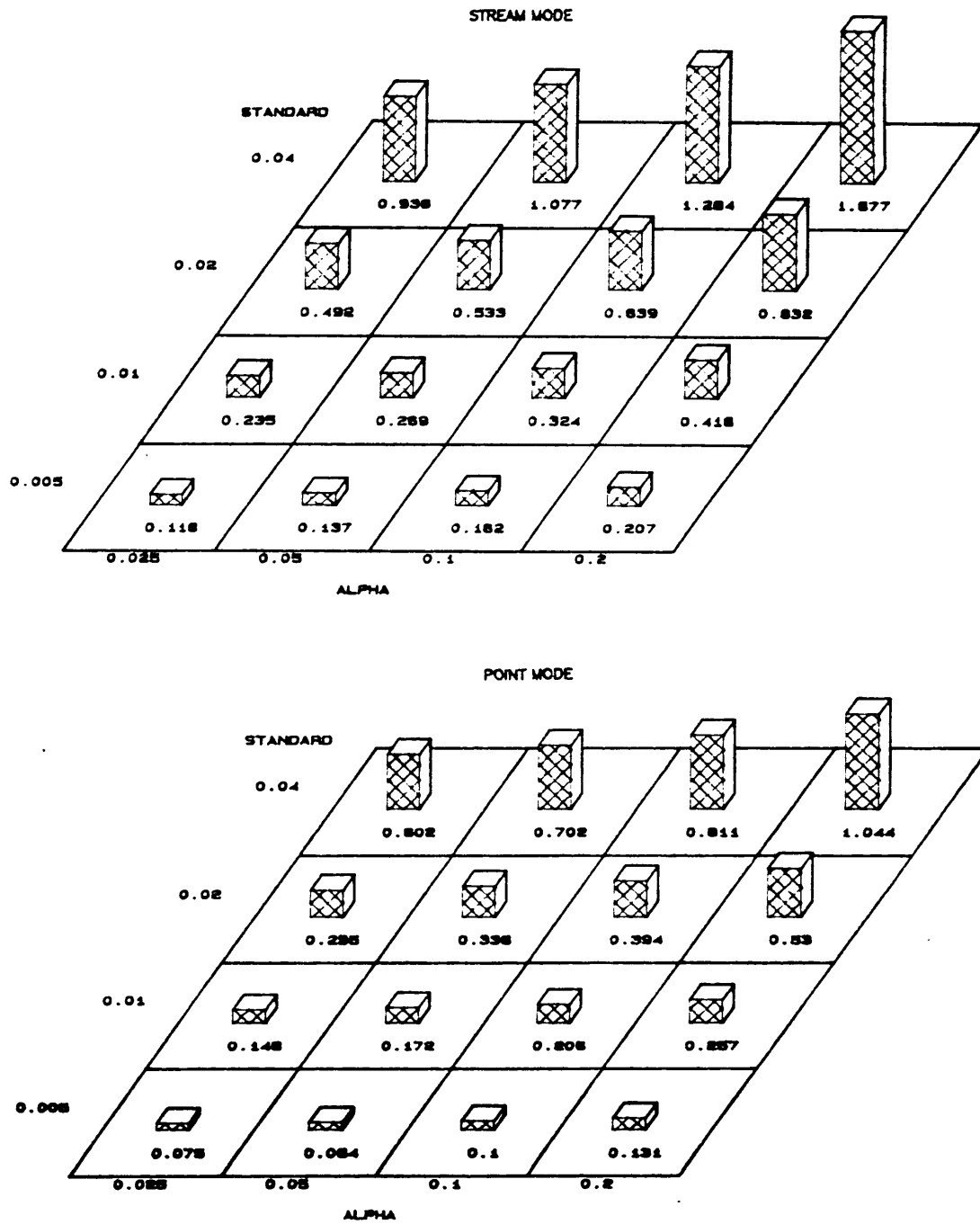


Figure 12: Average standard deviations of polygon area as percent of actual acres for a) stream mode, and b) point mode.

Table 9: Average bias of line length in stream mode simulations a) bias as actual feet, and b) bias as percent of actual feet.

Average Bias of Line Length (feet)

		Stream Mode			
		Alpha level			
		0.025	0.05	0.10	.20
Standard (map inches)	0.005	-9.080	-12.270	-17.500	-28.680
	0.010	-36.880	-48.390	-68.300	-113.520
	0.020	-145.850	-188.870	-264.080	-431.530
	0.040	-562.120	-720.930	-986.620	-1558.890

Average Bias of Line Length (% of actual feet)

		Stream Mode			
		Alpha level			
		0.025	0.05	0.10	0.20
Standard (map inches)	0.005	-0.066	-0.086	-0.124	-0.206
	0.010	-0.264	-0.348	-0.486	-0.813
	0.020	-1.039	-1.353	-1.896	-3.077
	0.040	-3.988	-5.153	-7.079	-11.119

Table 10: Average bias of line length in point mode simulations a) bias as actual feet, and b) bias as percent of actual feet.

Average Bias of Line Length (feet)

Point Mode

Alpha level

		0.025	0.05	0.10	.20
Standard (map inches)	0.005	-2.670	-3.660	-4.840	-7.780
	0.010	-10.110	-13.310	-19.160	-32.450
	0.020	-41.020	-54.370	-77.790	-129.580
	0.040	-167.890	-224.380	-314.510	-529.410

Average Bias of Line Length (% of actual feet)

Point Mode

Alpha level

		0.025	0.05	0.10	0.20
Standard (map inches)	0.005	-0.021	-0.029	-0.037	-0.059
	0.010	-0.076	-0.100	-0.145	-0.248
	0.020	-0.318	-0.416	-0.590	-0.989
	0.040	-1.273	-1.720	-2.395	-4.046

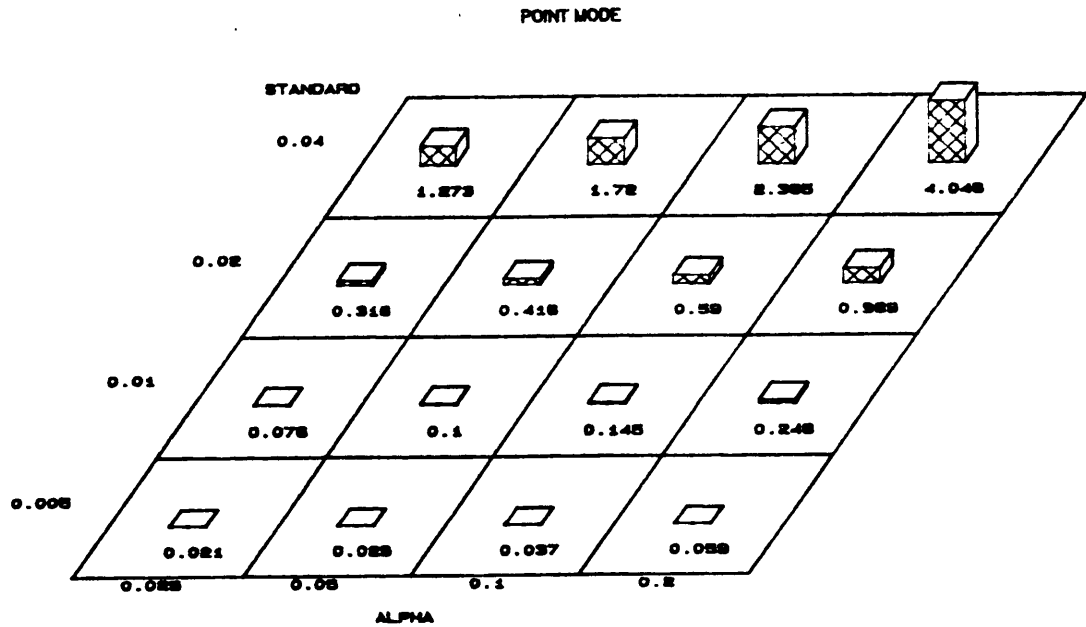
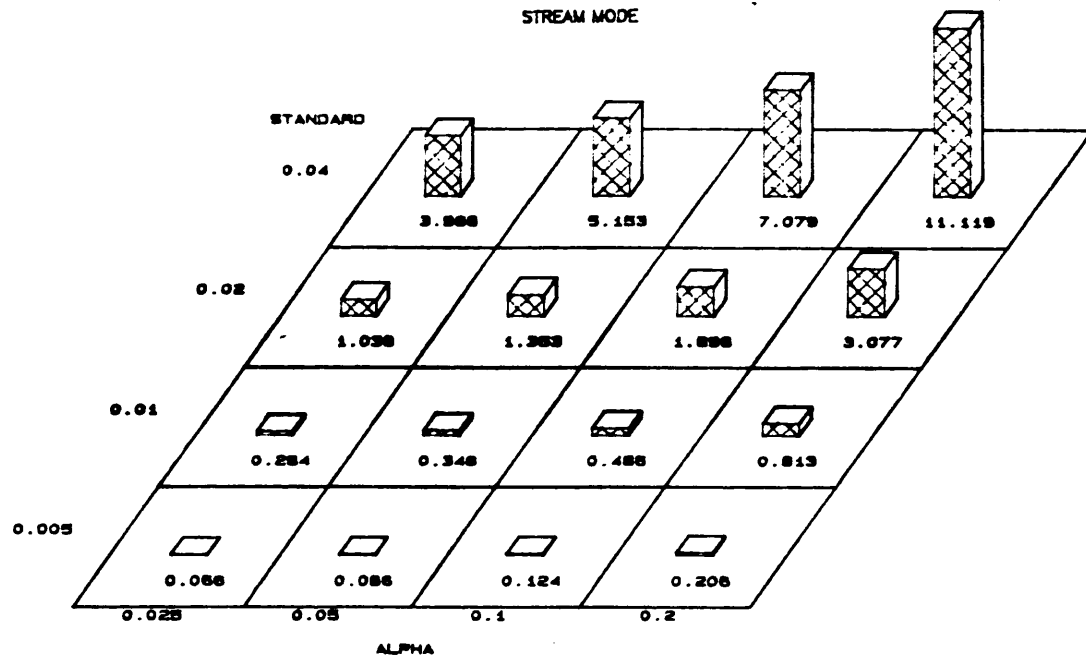


Figure 13: Average bias of line length as percent of actual feet for a) stream mode, and b) point mode.

The phenomenon of the quadrupling bias can be explained mathematically. The digitizing accuracy standard and associated error influenced line lengths can be diagrammed as shown in Figure 14.

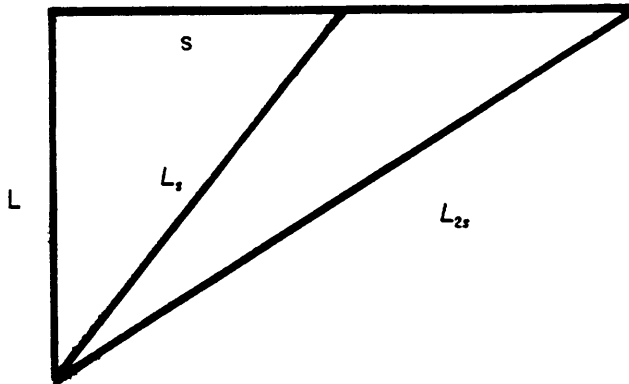


Figure 14: Schematic of accuracy standards and error influenced line lengths

where

S = accuracy standard bandwidth

L = true length of line

L_s = error influenced version of line at first accuracy standard

L_{2s} = error influenced version of line at doubled accuracy standard

Using the Pythagorean theorem, L_s and L_{2s} can be represented as

$$L_s^2 = L^2 + s^2$$

$$L_{2s}^2 = L^2 + 4s^2$$

$$\text{Bias at } s = L - L_s$$

$$\text{Bias as } 2s = L - L_{2s}$$

Therefore, the ratio between the two biases is

$$\frac{L - L_{2s}}{L - L_s} = \frac{L - \sqrt{L^2 + 4s^2}}{L - \sqrt{L^2 + s^2}}$$

This expression can be expanded using binomial expansion which yielded a ratio of 4. So by doubling the accuracy standard bandwidth, the bias quadrupled.

In stream mode, the bias of line length was always negative, indicating that the error influenced version of the line was always longer than the actual length. This was expected since deviations from the correct line increase cumulative length except in rare circumstances. These rare circumstances sometimes occurred in point mode simulation. Occasionally, when a particular feature had a positive bias, indicating an underestimation of the actual line length, the positive bias was always so small that the mean of the 25 features was still negative. This underestimation of the actual length only happened when a very small number of points (< 10) were used to digitize the feature. With only a few points, it was possible to digitize a line which was shorter than the actual line. Once a line had more than 10 points, this underestimation never occurred.

Table 11 shows the average standard deviations for line length in stream mode simulation. The values ranged from .089 percent of actual length to 4.670 percent of actual length. The trend across accuracy standards was not as clear as in previous cases. At the smaller accuracy standards, a doubling of the accuracy standard bandwidth produced an increase in standard deviation of approximately 2.5 times, but at the larger end of the accuracy standards, a doubling produced an increase in standard deviation of three times. Change in standard deviation across the alpha levels as alpha level doubled was approximately 1.2 to 1.4 times. The results from point mode simulation, shown in Table 12, indicated a rough doubling of the standard deviation as accuracy standard bandwidth doubled. The values ranged from .064 percent of actual length to 1.455 percent of actual length. As alpha level doubled, standard deviation increased approximately at the rate of 1.25 for the point mode simulations. It was expected that standard deviation of line length would double as accuracy standard bandwidth doubled as in the case of polygon area. However, another factor, such as line complexity, may have influenced the standard deviation. Regardless, the precision of line length was less at the less stringent accuracy standards. The difference in magnitude between stream mode and point mode was not a constant factor. At the .005 accuracy standard, stream mode was approximately 1.5 times larger than point mode, while at the .04 standard, stream mode was

Table 11: Average standard deviation of line length for stream mode simulations a) standard deviation as actual feet, and b) standard deviation as percent of actual feet.

Average Std. Dev. of Line Length (feet)

		Stream Mode			
		Alpha level			
		0.025	0.05	0.10	.20
Standard (map inches)	0.005	11.782	13.956	16.820	23.755
	0.010	27.780	33.585	43.919	65.063
	0.020	79.903	98.954	132.524	205.691
	0.040	258.409	319.288	432.669	650.480

Average Std. Dev. of Line Length (% of actual feet)

		Stream Mode			
		Alpha level			
		0.025	0.05	0.10	0.20
Standard (map inches)	0.005	0.089	0.105	0.127	0.179
	0.010	0.208	0.249	0.326	0.477
	0.020	0.588	0.720	0.965	1.498
	0.040	1.869	2.322	3.115	4.670

Table 12: Average standard deviation of line length for point mode simulations a) standard deviation as actual feet, and b) standard deviation as percent of actual feet.

Average Std. Dev. of Line Length (feet)

Point Mode

Alpha level

		0.025	0.05	0.10	.20
Standard (map inches)	0.005	8.008	9.196	10.987	14.046
	0.010	16.004	18.489	22.402	29.455
	0.020	33.666	40.128	50.212	67.642
	0.040	78.732	98.357	125.105	182.617

Average Std. Dev. of Line Length (% of actual feet)

Point Mode

Alpha level

		0.025	0.05	0.10	0.20
Standard (map inches)	0.005	0.064	0.072	0.087	0.110
	0.010	0.126	0.146	0.176	0.235
	0.020	0.269	0.319	0.402	0.540
	0.040	0.634	0.791	0.995	1.455

3.0 times larger. Figure 15 shows a comparison between the standard deviations in line length for stream mode and point mode data. The bias of polygon area displayed no trends since the bias was near zero at all accuracy standards. The variation in polygon area increased as accuracy standards became less stringent with a doubling of standard deviation occurring as accuracy standard bandwidth doubled. Values for standard deviations of polygon area ranged from .118 percent to 1.677 percent in stream mode and .075 percent to 1.044 percent in point mode. The biases and variances in line length both increased as the accuracy standard increased. A doubling of the accuracy standard bandwidth caused a quadrupling of the line length bias and a doubling to tripling of the line length standard deviation. The magnitude of stream mode biases and standard deviations was always larger than the corresponding point mode values. The magnitude of the effect of the digitizing errors was not extremely large for point mode or stream mode especially when the error was viewed as a percentage of actual area or actual line length.

Trends Across Feature Size

Each of the plots which appeared in this section reflected results of simulations for all four accuracy standard bandwidths at the .025 alpha level. Plots at the other three alpha level always displayed the same trends as their corresponding .025 alpha level plots with only a change in magnitude. An entire set of plots was collected and presented in appendix D.

Bias of polygon area:

Bias of polygon area for stream mode simulations was plotted against polygon area to reveal a random pattern centered around zero as shown in Figure 16. The same random pattern appeared when the point mode data were plotted. These plots showed no change in bias as feature size increased, but the amount of variation from zero did increase with wider accuracy standards. The point mode data were also plotted against number of points per

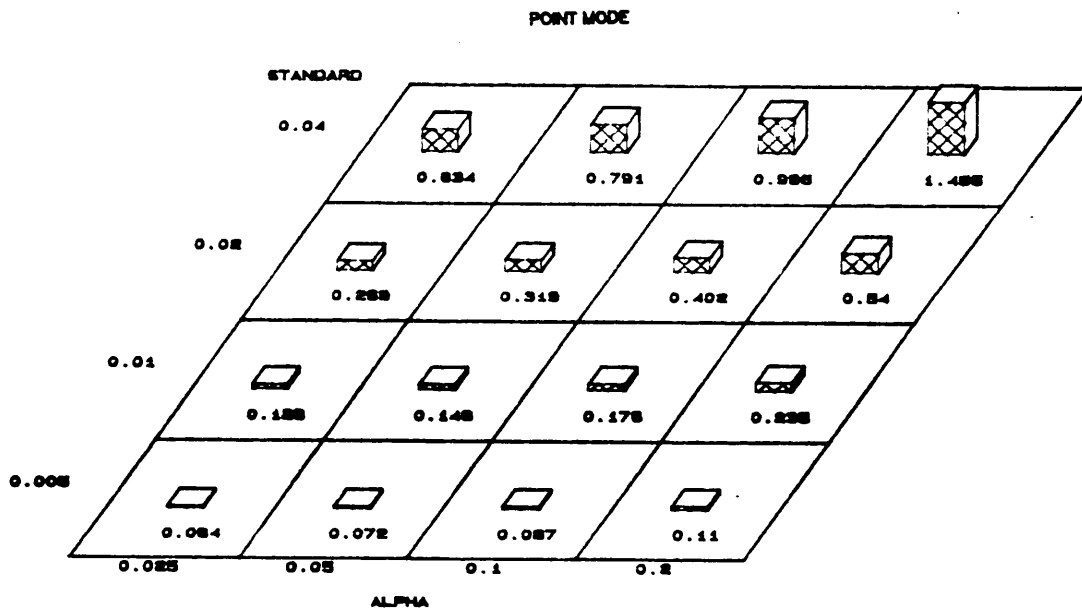
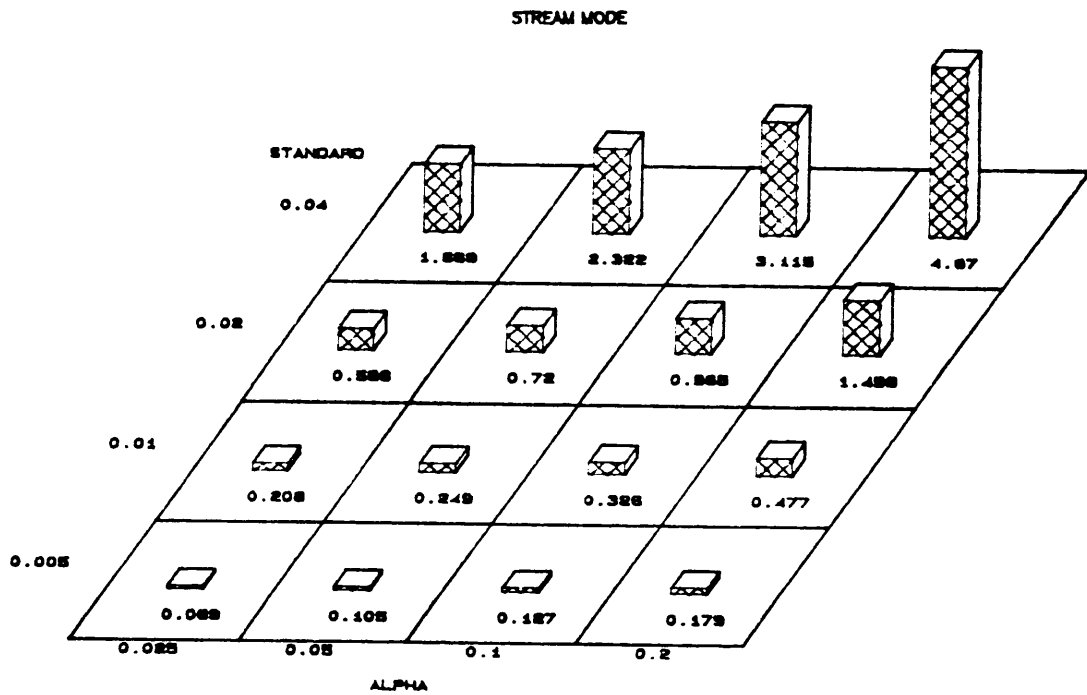


Figure 15: Average standard deviations of line length as a percent of actual feet for a) stream mode, and b) point mode.

BIAS VS. POLYGON AREA

Stream data, Alpha = .025

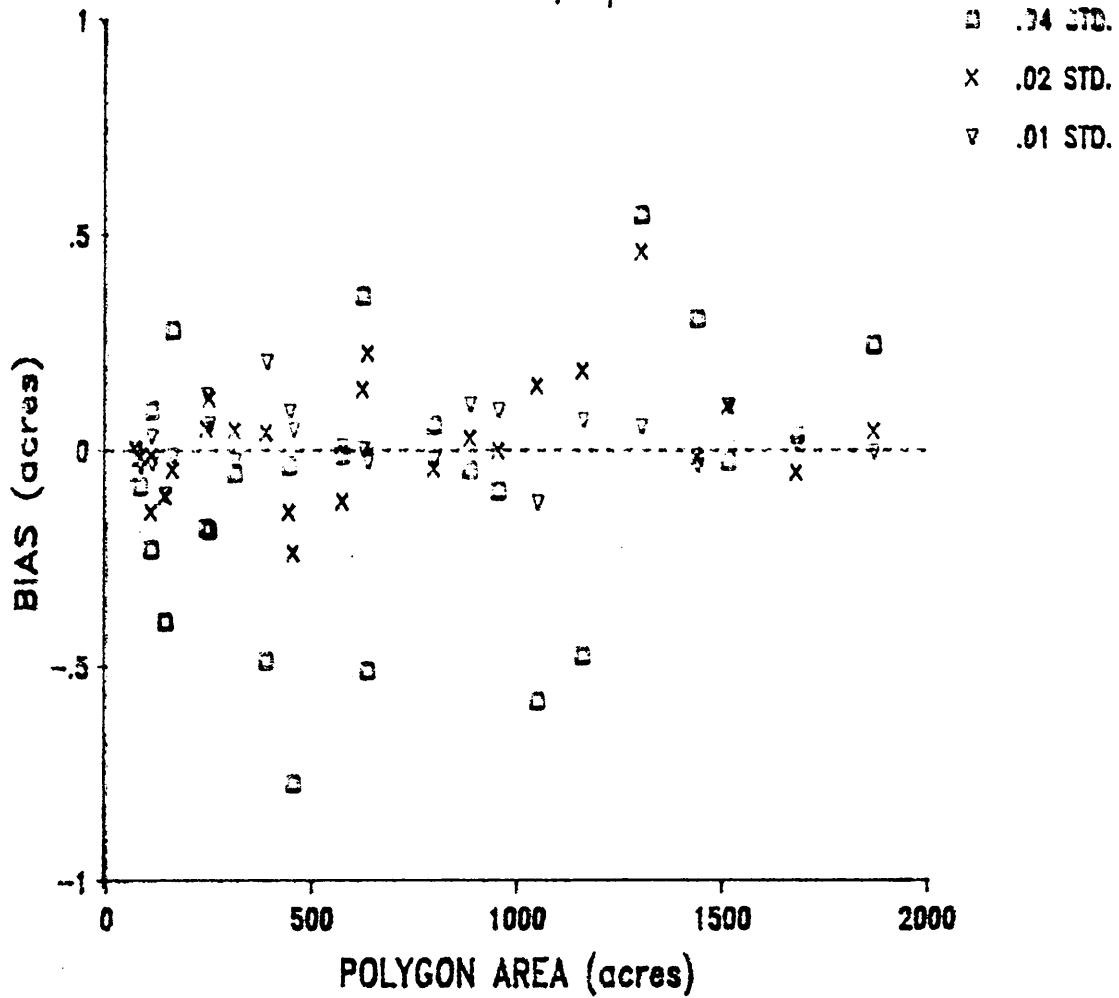


Figure 16: Bias of polygon area plotted against polygon area for stream mode simulation at alpha level .025.

feature. These plots were no different from the plots using polygon area. Pearson product-moment correlation coefficients were calculated for the variable combinations. Coefficients near 1.0 indicated strong linear correlation, while correlation coefficients less than .5 generally indicated the lack of a linear relationship. An analysis of correlation coefficients showed very low correlation between bias and polygon area and very low correlation between bias and number of points per feature. For the stream mode results, there was a strong positive correlation between size of the feature in acres and the number of points per feature. These results appeared reasonable since the bias of average polygon area was shown to be approximately zero in the previous section. The size of the polygon had no effect on the bias because the errors made in digitizing a polygon balanced themselves in such a way that the net error, or bias, was zero. Stream mode and point mode exhibited no differences in their bias across polygon size. Neither the method of digitizing, nor the size of the polygon in acres, nor the number of points used to digitize the polygon affected the tendency for bias to be close to zero.

A plot of relative bias against polygon area for stream mode data is shown in Figure 17. The data produced a funnel shaped plot with the widest portion of the funnel occurring at the smallest polygons. For small polygons, bias was large relative to the size of the polygon with values as large as .2 percent of polygon area, and the relative bias decreased as the polygon size increased. A plot of the point mode data revealed a similar trend only with a lesser degree of magnitude than the stream mode plot, note change of scales in plots (Figure 18). Correlation coefficients of .4 and .6 showed that after performing a log transformation on the relative bias, relative bias was well correlated with polygon size and with number of points.

Since the absolute bias was constant as feature size changed, it followed naturally that relative bias decreased as feature size increased. If all polygons had the same amount of error in absolute terms, then that error represented a larger portion of the total area of a small polygon than it did for a large polygon. Even though this trend was present in the relative bias,

RELATIVE BIAS VS. POLYGON AREA

Point data, Alpha = .025

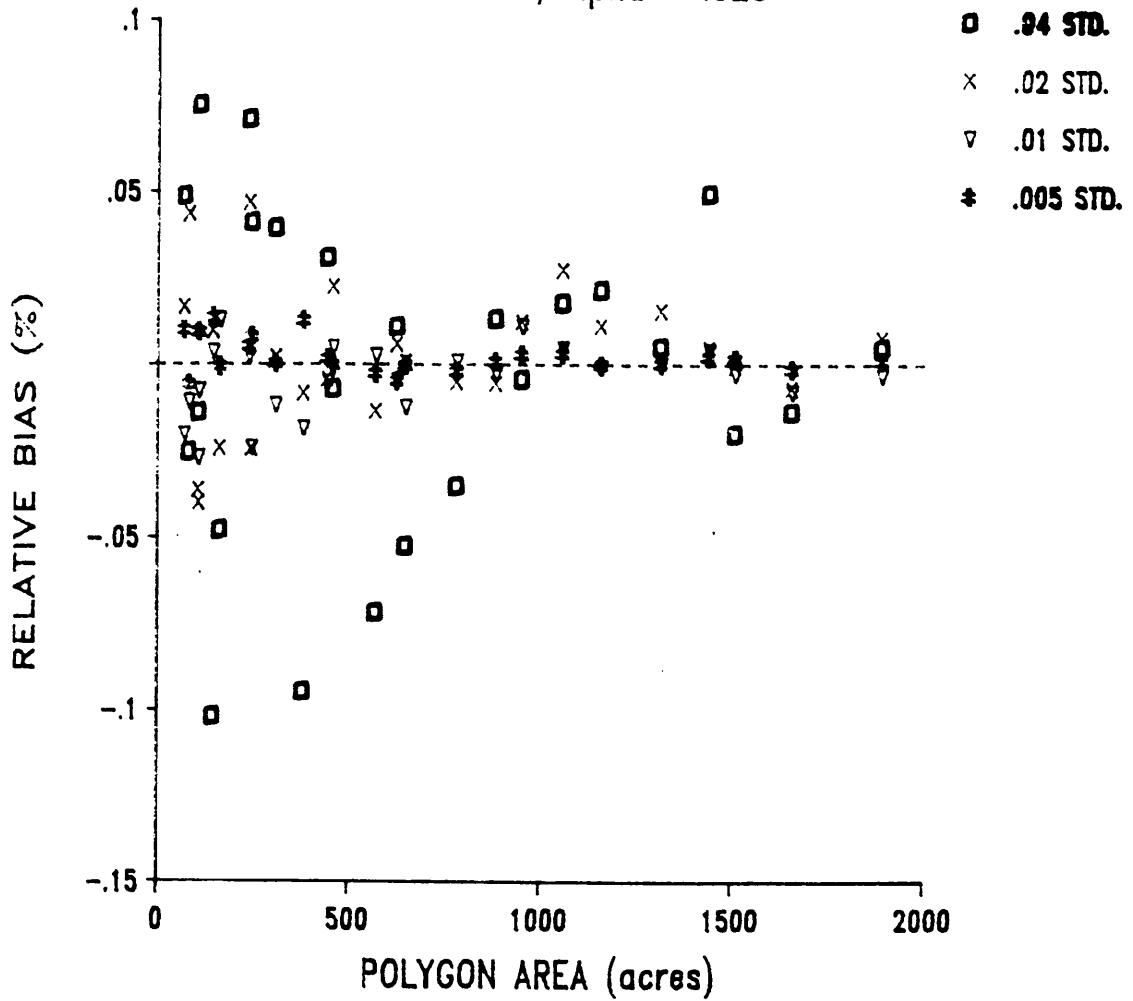


Figure 18: Relative bias of polygon area plotted against polygon area for point mode simulations at alpha .025.

the magnitude of this error must be kept in perspective. The relative biases were less than .3 percent in stream mode and less than .1 percent in point mode.

Variation in polygon area:

When the standard deviation of polygon area was plotted against polygon size for both point mode and stream mode simulations, a strong positive linear trend was revealed (Figure 19). As polygon size increased, the standard deviation increased at a constant rate. The stream mode plot showed strong linearity and distinct groupings by accuracy standards while the point mode plots were slightly less linear and the accuracy standards tended to overlap somewhat. The point mode and stream mode plots followed the same trends, but the magnitude of the stream mode standard deviations was 1.5 times that of the point mode standard deviations. The stream mode data showed high positive correlation coefficients between standard deviation and polygon size and between standard deviation and number of points per polygon. In point mode, polygon size was highly correlated with standard deviation, but number of points was not strongly correlated with standard deviation. This resulted because there was not a strong correlation between polygon size and the number of points when simulating digitizing in point mode. There was a strong correlation between these two variables when stream mode digitizing was simulated.

Although the long run average, or bias, in polygon error was zero for any size polygon, the amount a particular simulated polygon could vary from the true polygon acreage increased as feature size increased. Therefore, larger polygons had less precision than smaller polygons in terms of variance. For a particular accuracy standard, the rate of increase was fairly constant, but as the accuracy standard bandwidth increased, the rate increased. A wider accuracy standard produced a more rapid increase in the variation of polygon error as feature size increased than did a more stringent accuracy standard. Also wider accuracy standards permitted a wider range of variation than did tight standards.

STD DEV VS. POLYGON AREA

Stream data, Alpha = .025

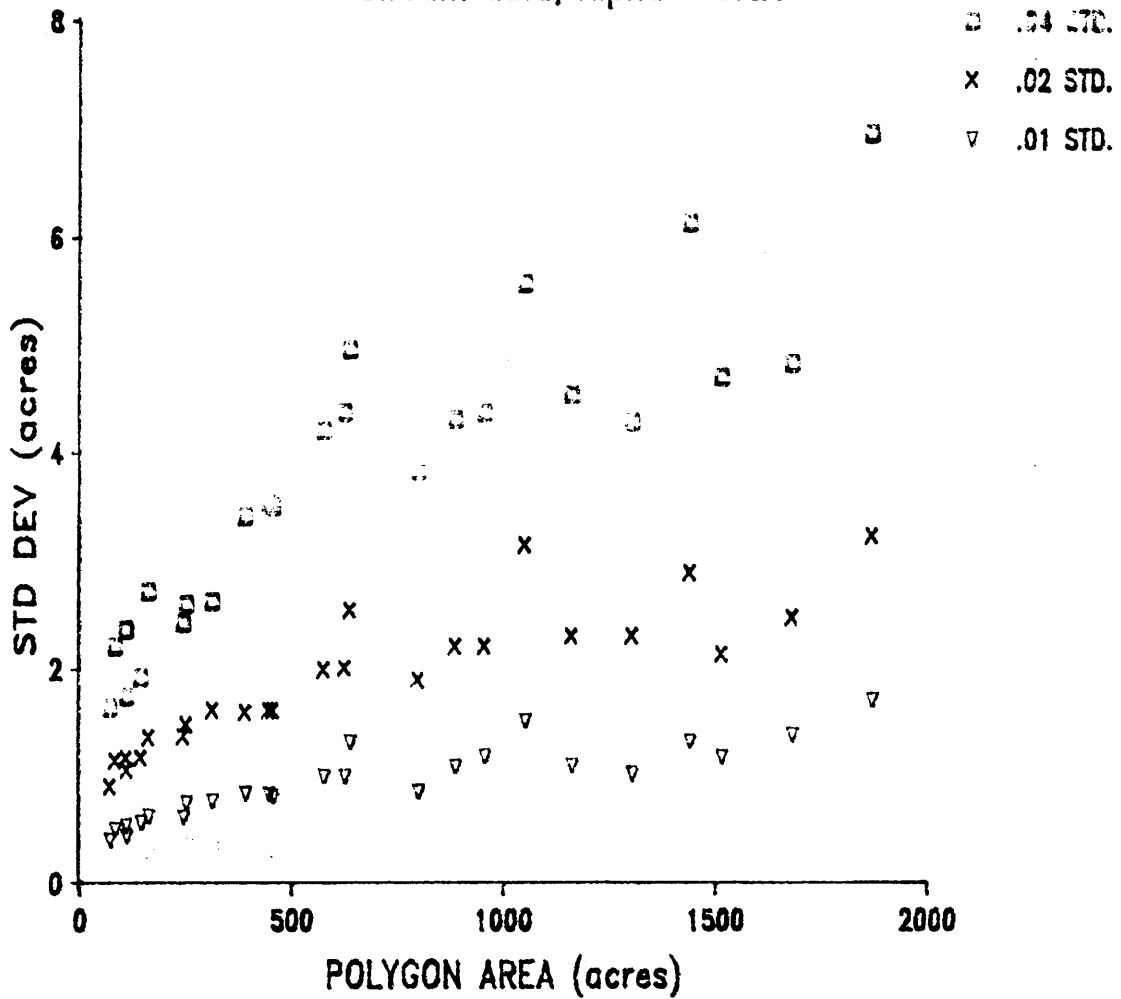


Figure 19: Standard deviation of polygon area plotted against polygon size for stream mode simulations at alpha .025.

A plot of relative standard deviation of polygon area over polygon area for stream mode is shown in Figure 20. A distinct J-curve trend was evident from this plot such that smaller polygons had considerably larger relative standard deviation of polygon area than did larger polygons. The four accuracy standards were quite distinct in the stream mode plot. A plot of the relative standard deviation for point mode simulations also revealed a J-curve plot although not as distinct as that of the stream mode data (Figure 21). In point mode, the accuracy standards tended to overlap, just as they did in the standard deviation plots. By using an inverse transformation on the relative standard deviation, a strong negative correlation coefficient of $-.9$ was calculated for relative standard deviation and polygon size in stream mode and point mode. Although the standard deviation increased as polygon size increased, it did not increase as rapidly as polygon size increased. Therefore, the J-curve plot resulted when relative standard deviation was plotted against polygon size. As polygon size increased, the standard deviation, or variation in polygon area, represented a smaller percentage of the overall size of the polygon. The error was highest for a few of the smallest polygons with as much as 2.5 percent of polygon area for stream mode. The variance then rapidly diminished and appeared to stabilize at approximately a polygon size of 1000 acres, which at a 1:24000 scale is approximately 10 square map inches. For polygons larger than 1000 acres, the relative standard deviation was less than .5 percent of total area.

Smaller polygons tended to have more bias and more variance relative to their size than larger polygons. Therefore, when digitizing small polygons, more accurate digitizing standards should be adhered to in order to compensate for the greater potential for error. The number of points in a polygon produced responses similar to those produced by polygon size in acres since polygon size and number of points are strongly correlated in stream mode digitizing. The point mode results always had trends similar to stream mode, except point mode tended to produce somewhat less distinct trends which resulted from the more subjective style of recording points when digitizing in point mode.

RELATIVE STD DEV VS. POLYGON AREA

Stream data, Alpha = .025

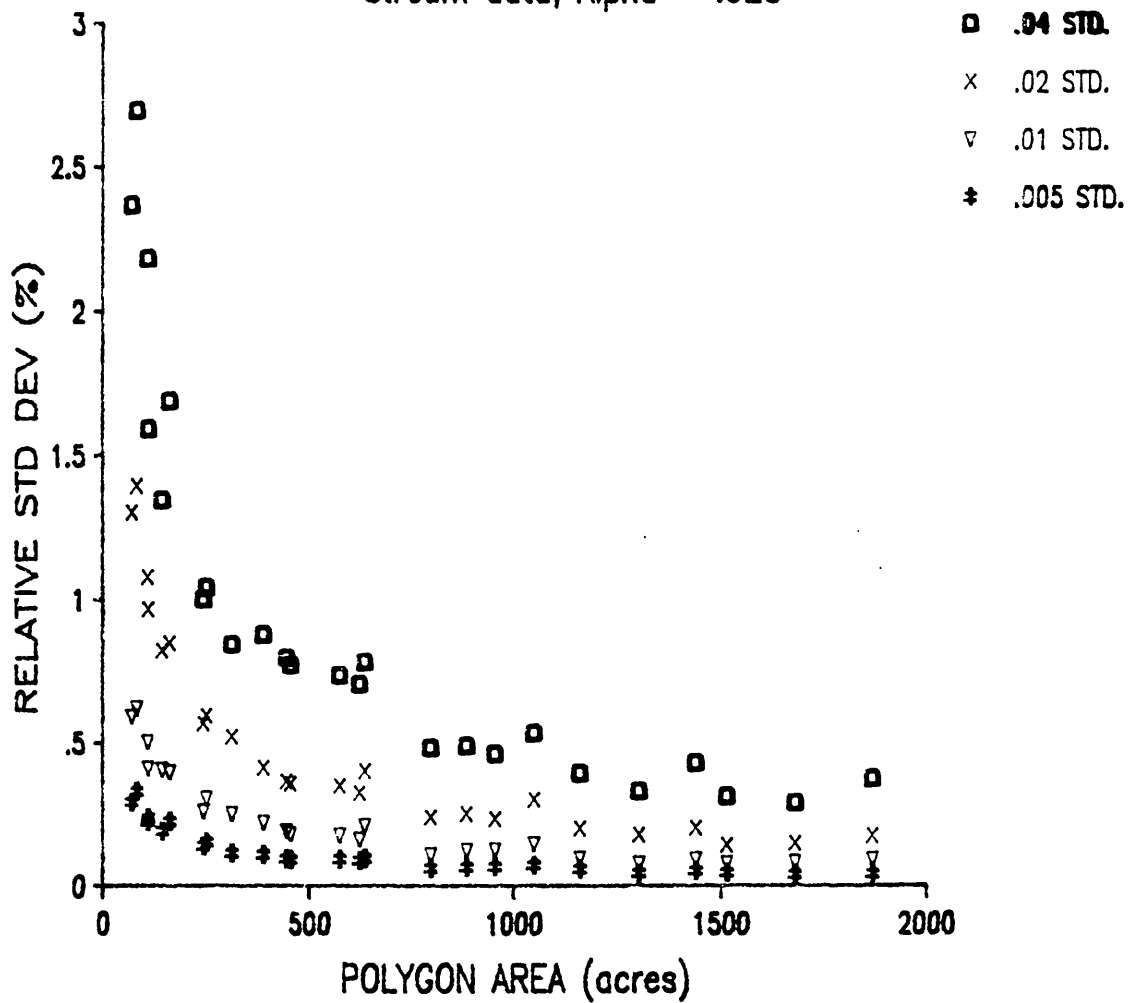


Figure 20: Relative standard deviation of polygon area plotted against polygon area for stream mode simulations at alpha .025.

RELATIVE STD DEV VS. POLYGON AREA

Point data, Alpha = .025

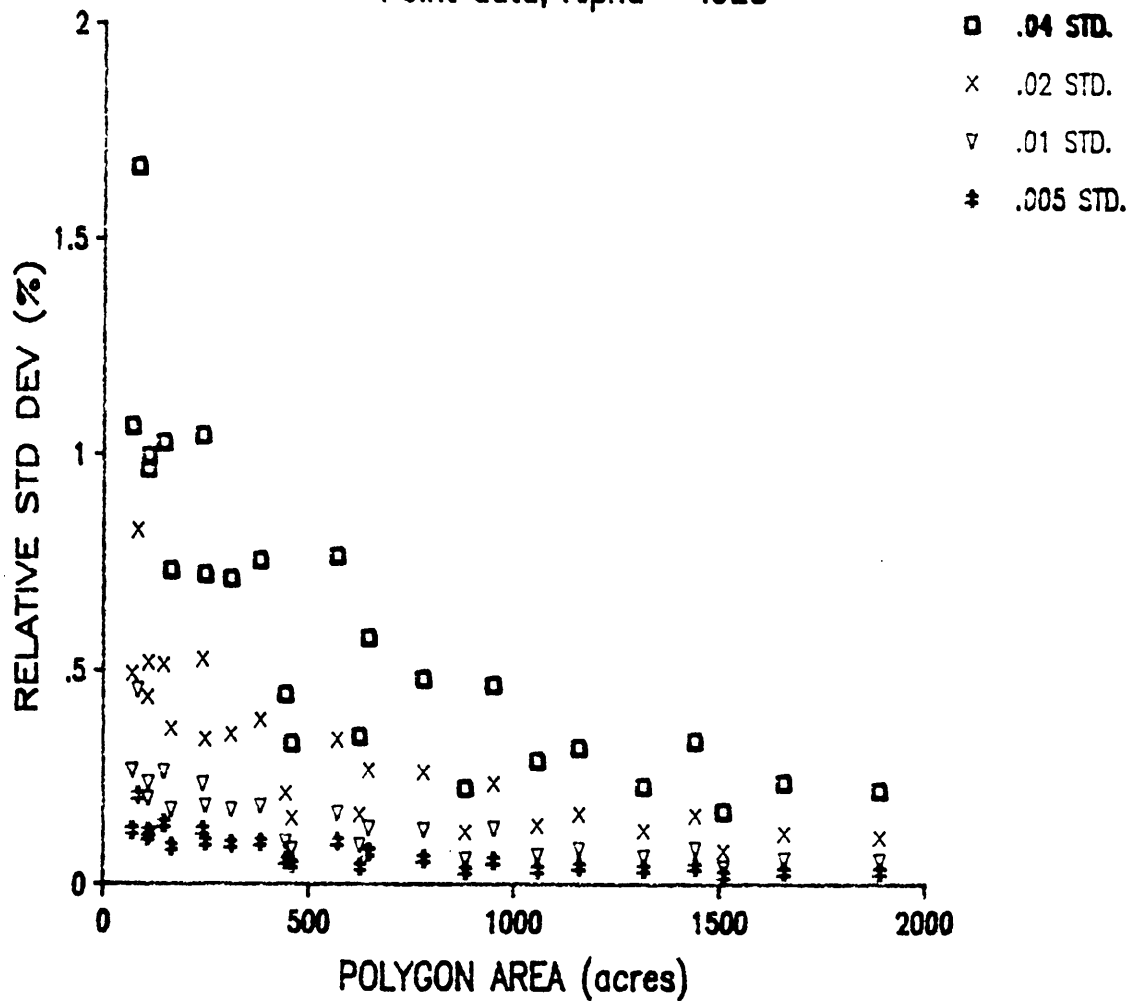


Figure 21: Relative standard deviation of polygon area plotted against polygon area for point mode simulations at alpha .025.

Bias of line length:

Bias for line length was plotted against line length, and the resulting plot for stream mode is shown in Figure 22. Obvious strong, positive linear trends are evident between increasing bias and increasing line length. The longer a line was, the more biased it was in absolute terms. The slope of the plotted data points was fairly steep which indicated a rapid increase in bias across increasing feature size. Wider accuracy standards had steeper slopes than lower accuracy standards. A correlation coefficient of .92 was calculated for the correlation between bias in stream mode and line length. Point mode differed in that bias was not very well correlated with line length in feet. Figure 23 illustrates this lack of correlation in point mode.

Plots of bias against the number of points in a line produced some interesting results. For stream mode, the plot using number of points per line was almost exactly identical to the plot using line length. This observation was further supported by correlation coefficients of .94 for correlation between bias and number of points and .99 for correlation between number of points against line length. Line length and number of points were so strongly correlated in stream mode digitizing that they essentially produced the same trends. Plotting bias against number of points for the point mode simulations revealed some positive linear correlation not seen in the previous plot of point mode using line length (Figure 24). The correlation coefficient between bias and number of points for the point mode data was .8. In contrast to stream mode, point mode digitizing did not produce lines with the number of points being well correlated with line length.

When digitizing in stream mode, the bias increased as the line length increased. Error in line length was cumulative for there was no compensating effect of errors as in polygon areas; therefore, the longer a line was, the more error accumulated and produced a high bias. As discussed in an earlier section, the bias is almost always an overestimation of the true length because as a digitizing operator follows a line, the criss-cross pattern of digitizing error

BIAS VS. LINE LENGTH

Stream data, Alpha = .025

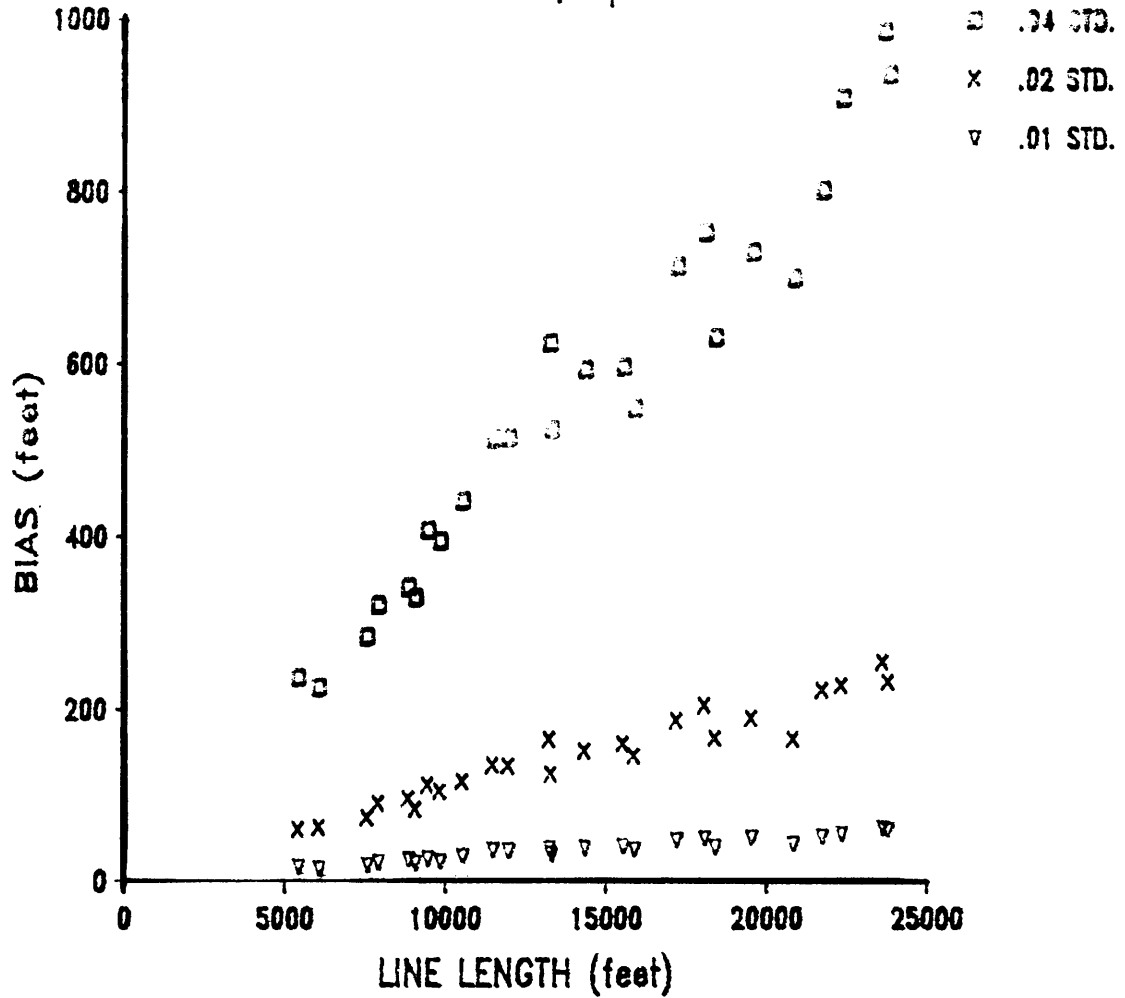


Figure 22: Bias of line length plotted against line length for stream mode simulations at alpha .025.

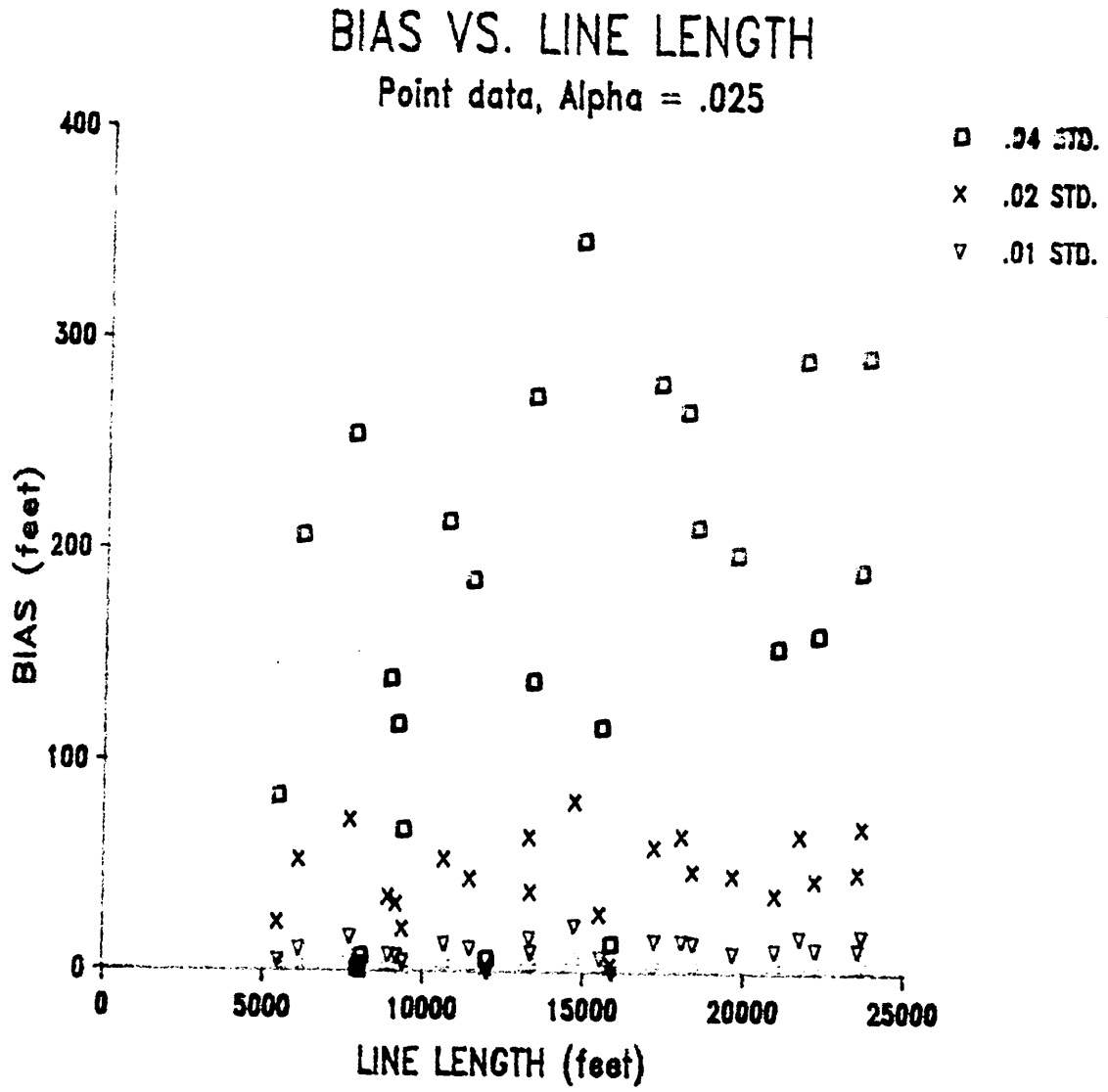


Figure 23: Bias of line length plotted against line length for point mode simulations at alpha .025.

BIAS VS. NUMBER OF POINTS

Point data, Alpha = .025

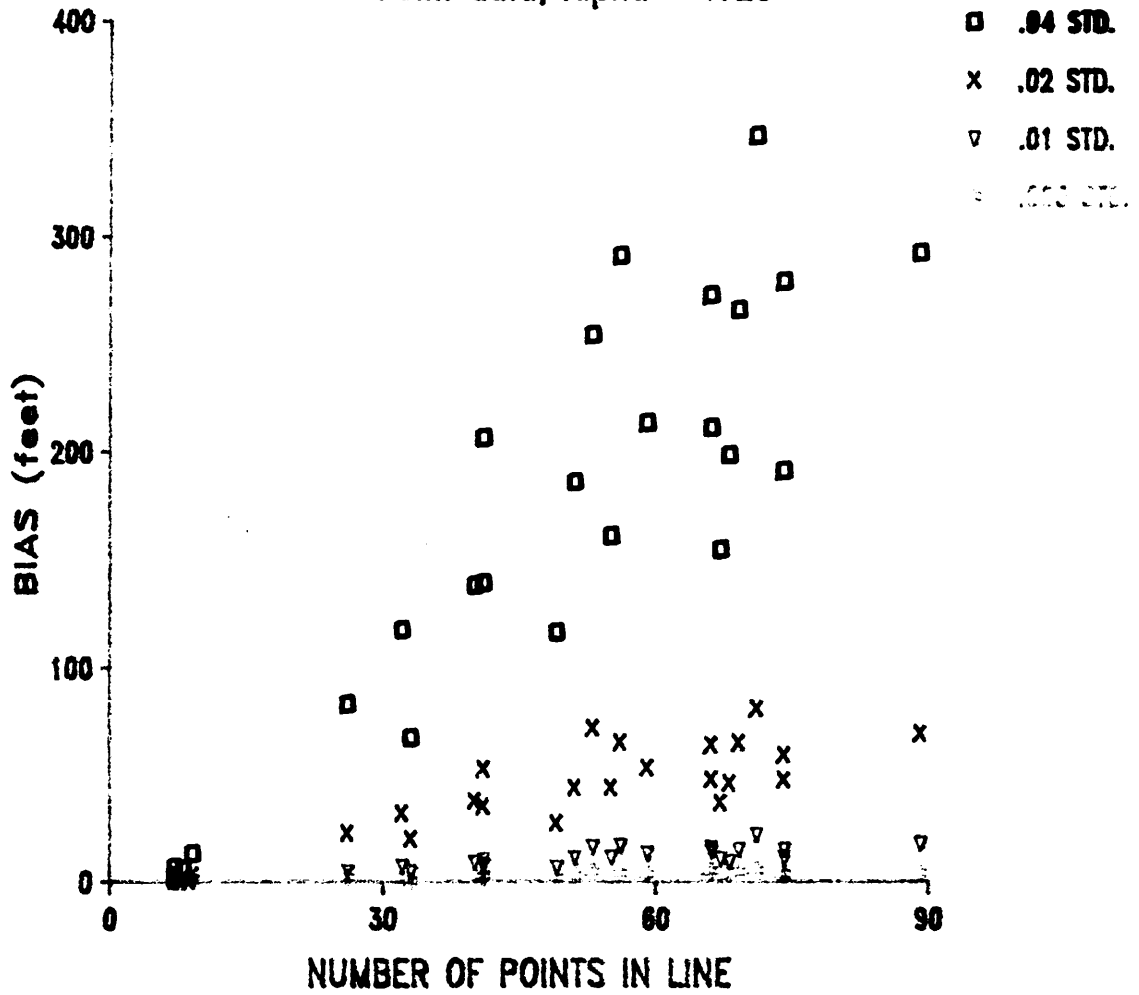


Figure 24: Bias of line length plotted against number of points per line for point mode simulations at alpha .025.

creates a longer version of the line. The number of points in a line digitized in stream mode exhibited a high positive correlation with line length, a result caused by the stream mode digitizing process where a point is recorded at equal intervals. The results from point mode simulations showed that bias was more related to the number of points in a line rather than the line's length. This appeared reasonable since the error was produced by the movement of the point itself, and not the distance between points. Therefore, error accumulates with the addition of each point and not the distance between the points. This statement may also be true of stream mode, but it is difficult to discern whether line length or number of points is the actual cause of the trend in bias since they themselves are so strongly correlated in the stream mode data.

Relative bias of line length was plotted against line length for stream mode data and is shown in Figure 25. The plots revealed that there was no change in relative bias as line length increased. This indicated that bias, as seen in Figure 22, increased at a rate equal to the increase in line length. Therefore, the bias for a short line as a percentage of its length was equal to the bias of a long line in terms of a percentage of its length. The amount of bias per unit length of line was always the same regardless of total line length.

As expected, a plot of relative bias against line length for the point mode simulations revealed no trend. The small amount of correlation which was present indicated a slight decrease in relative bias as line length increased. A somewhat horizontal plot similar to that achieved for the stream mode simulations was obtained by plotting relative bias against number of points per line for point mode (Figure 26). This supported the idea that bias in point mode was more related to number of points than to line length. An analysis of correlation coefficients showed that there was a very high correlation of .9 between relative bias and points per foot in the point mode data. Figure 27 displays a plot of relative bias against points per foot for point mode. In stream mode, points per foot is a constant, but in point mode the digitizing operator makes the decision of how many points per foot there will be, a decision

RELATIVE BIAS VS. LINE LENGTH

Stream data, Alpha = .025

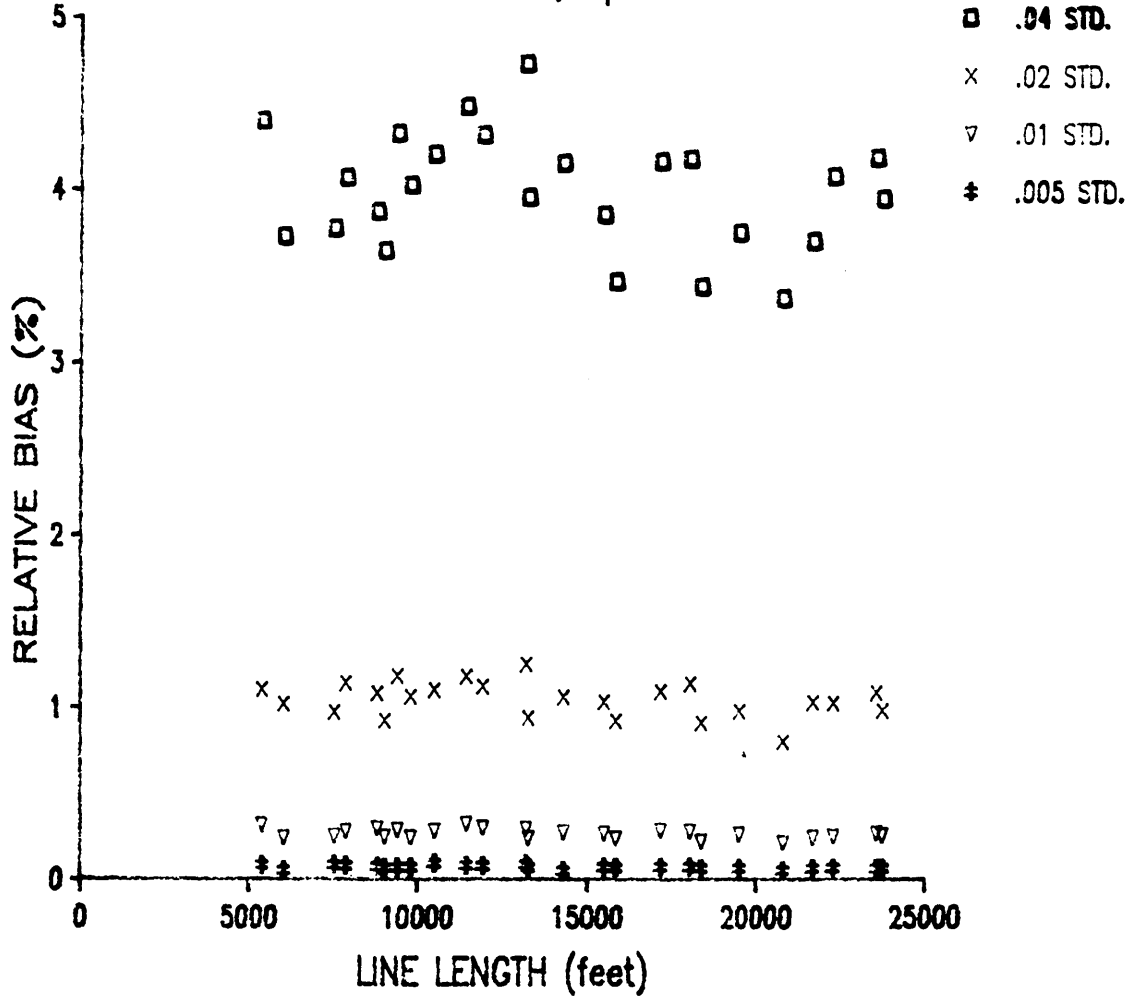


Figure 25: Relative bias of line length plotted against line length for stream mode simulations at alpha .025.

RELATIVE BIAS VS. NUMBER OF POINTS

Point data, Alpha = .025

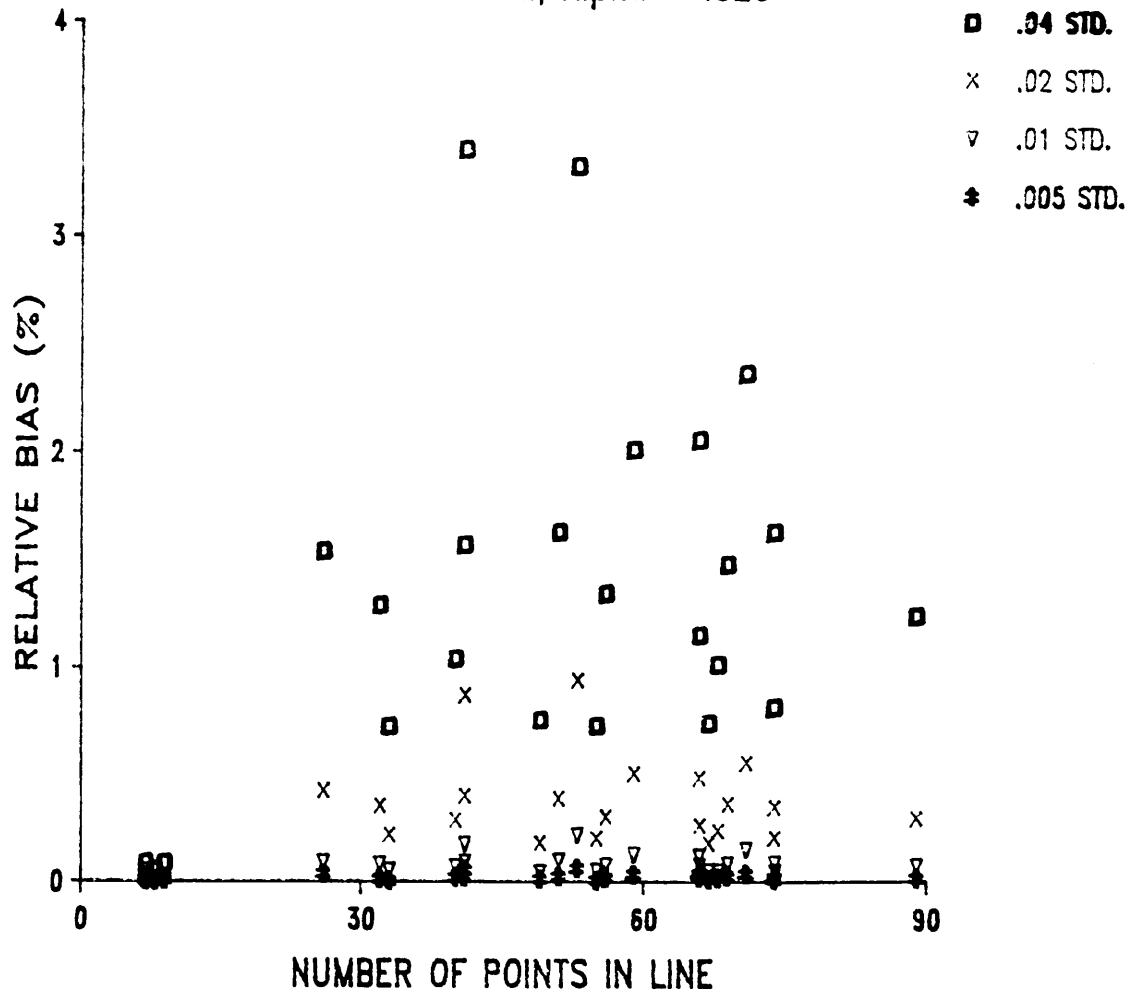


Figure 26: Relative bias of line length plotted against number of points per line for point mode simulations at alpha .025.

RELATIVE BIAS VS. POINTS PER FOOT

Point data, Alpha = .025

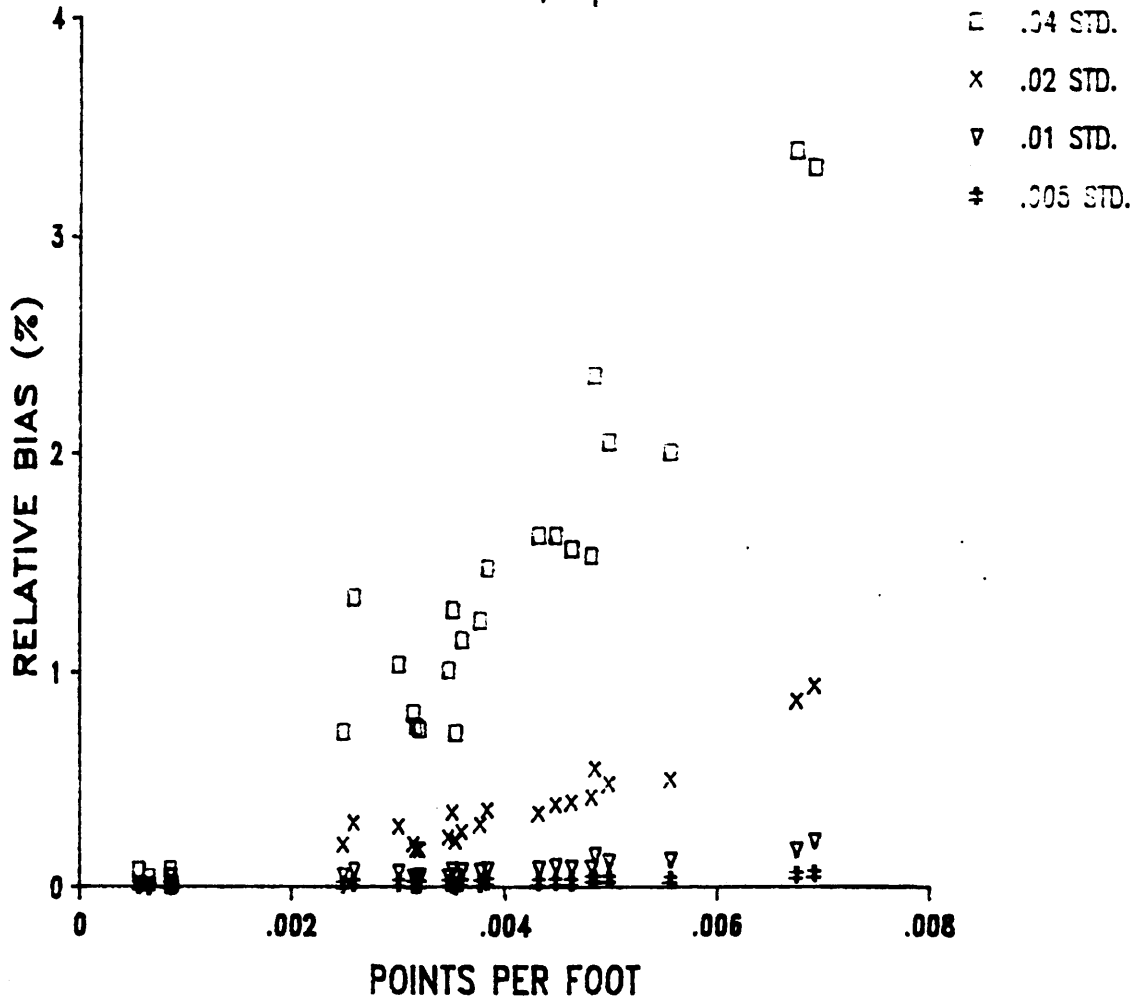


Figure 27: Relative bias of line length plotted against point per foot for point mode simulations at alpha .025.

usually influenced by the complexity of the line. A more complex line usually requires a more dense sampling of points to achieve an accurate representation of the line and hence results in more points per foot. Figure 27 indicates that as a line became more complex, bias as a percent of total line length increased. Figure 28 shows that as points per foot increased (i.e. line become complex), bias remained relatively constant.

The results of the bias in stream mode analysis were easy to interpret. The amount of bias was the same per unit length of line, and bias accumulated with increasing line length. The point mode results were more difficult to explain. The plot of bias to number of points indicated bias in point mode was more related to number of points rather than line length. However, the plots of bias against points per foot appeared to show that regardless of the number of points per foot, the bias was essentially the same which indicated a constant bias per unit length as in stream mode. Perhaps underlying this apparent disagreement in point mode results was an interaction between number of points in a line, line length, and line complexity. Further studies need to be done to examine the influence of line complexity when digitizing in point mode.

Variation in line length:

Standard deviation and relative standard deviation behave very similar to bias and relative bias. Figure 29 shows the linear increase in standard deviation as line length increased for the stream mode simulations. Again, the standard deviation increased at a rate equal to the increase in line length such that relative standard deviation plotted against line length exhibited the horizontal plot with a slight negative slope as shown in Figure 30. Therefore, variation in line length from the true length was constant per given unit of length, but it was cumulative over the total length.

Standard deviation for point mode also showed characteristics similar to the bias for point mode simulations. Again, there was more correlation between standard deviation and

BIAS VS. POINTS PER FOOT

Point data, Alpha = .025

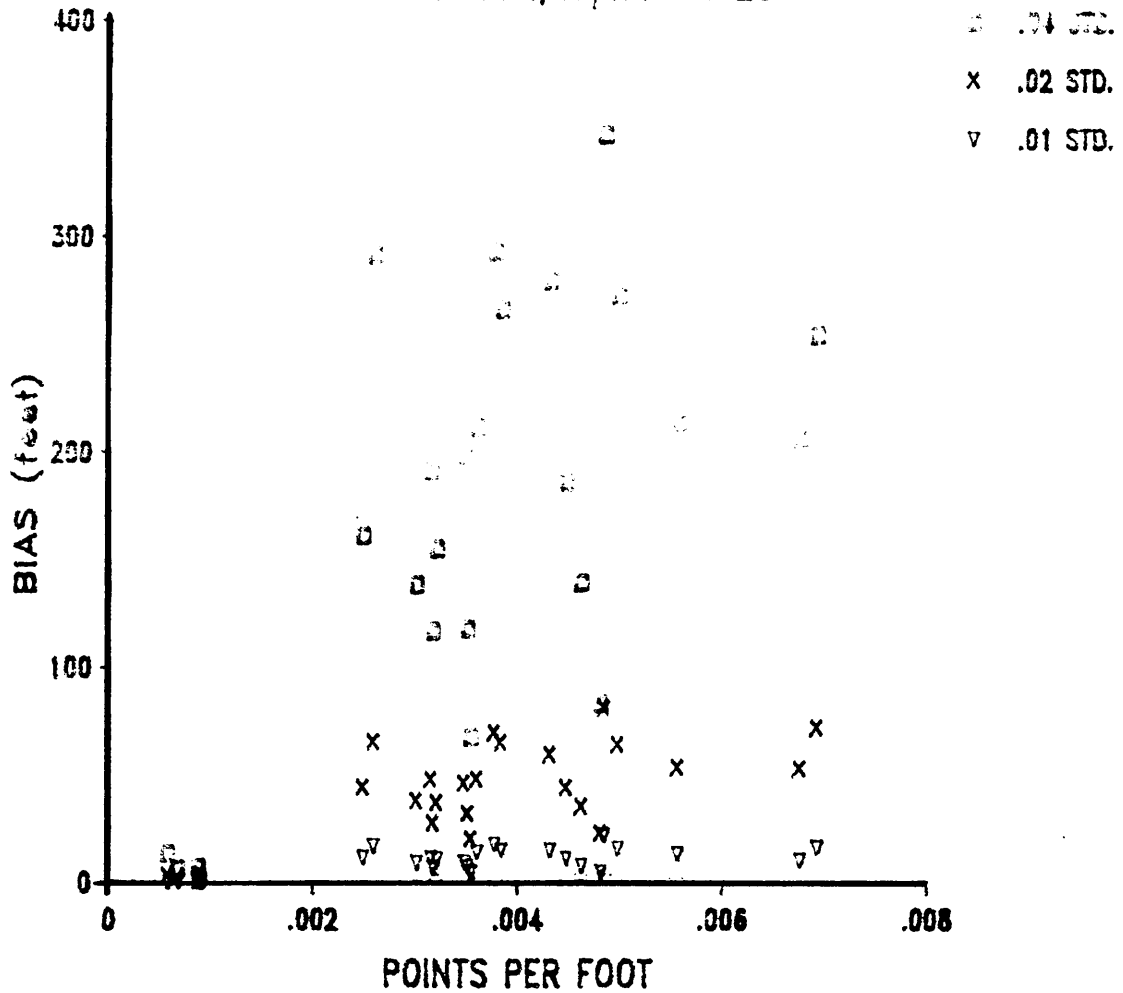


Figure 28: Bias of line length plotted against point per foot for point mode simulations at alpha .025.

STD DEV VS. LINE LENGTH

Stream data, Alpha = .025

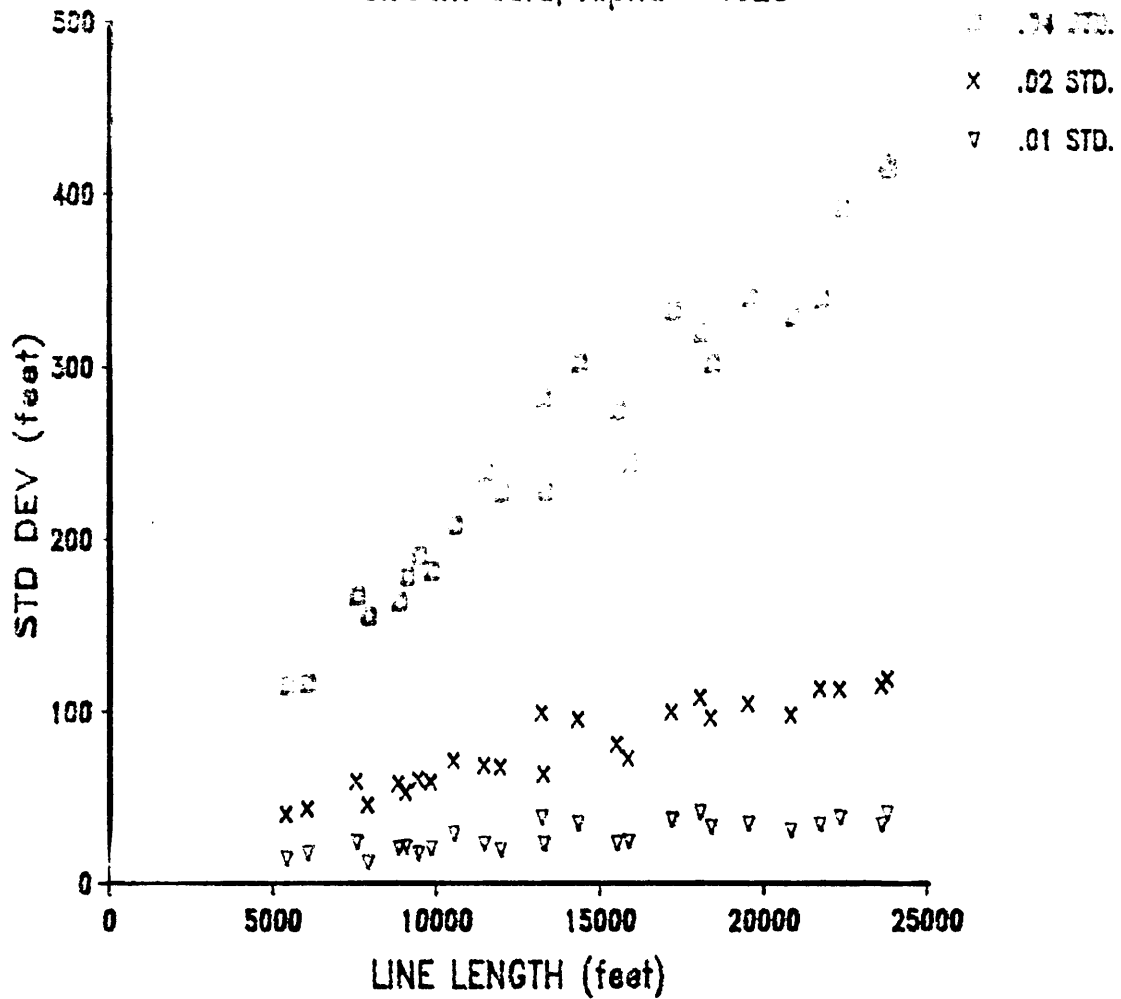


Figure 29: Standard deviation of line length plotted against line length for stream mode simulations at alpha .025.

RELATIVE STD DEV VS. LINE LENGTH

Stream data, Alpha = .025

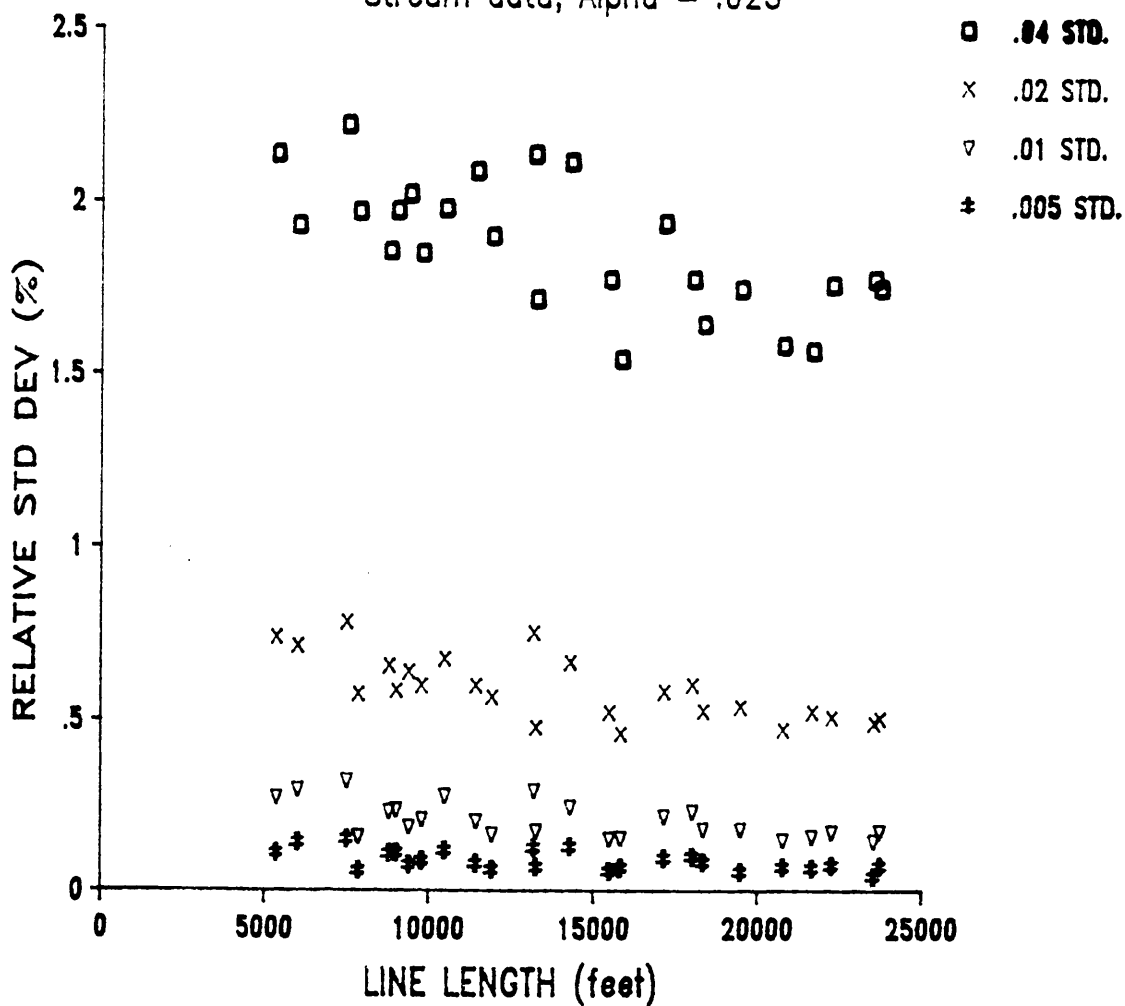


Figure 30: Relative standard deviation of line length plotted against line length for stream mode simulations at alpha .025.

number of points rather than actual line length. Figure 31 shows that standard deviation increased with increasing number of points. Relative standard deviation was roughly constant across number of points per line. For the case of standard deviation, the plots of points per foot showed no discernible trends. In summary of line length analysis, stream mode digitizing errors caused bias and standard deviation to increase with increasing line length (or number of points), but the bias and standard deviation as a percent of total line length remained constant across feature size change. In point mode, the effects of digitizing error appeared to be most strongly related to the number of points per line. Bias and standard deviation of line length exhibited relationships with number of points per line similar to those found in the stream mode data. Line complexity became a confounding factor in the analysis of point mode error trends and warrants further investigation.

STD DEV VS. NUMBER OF POINTS

Point data, Alpha = .025

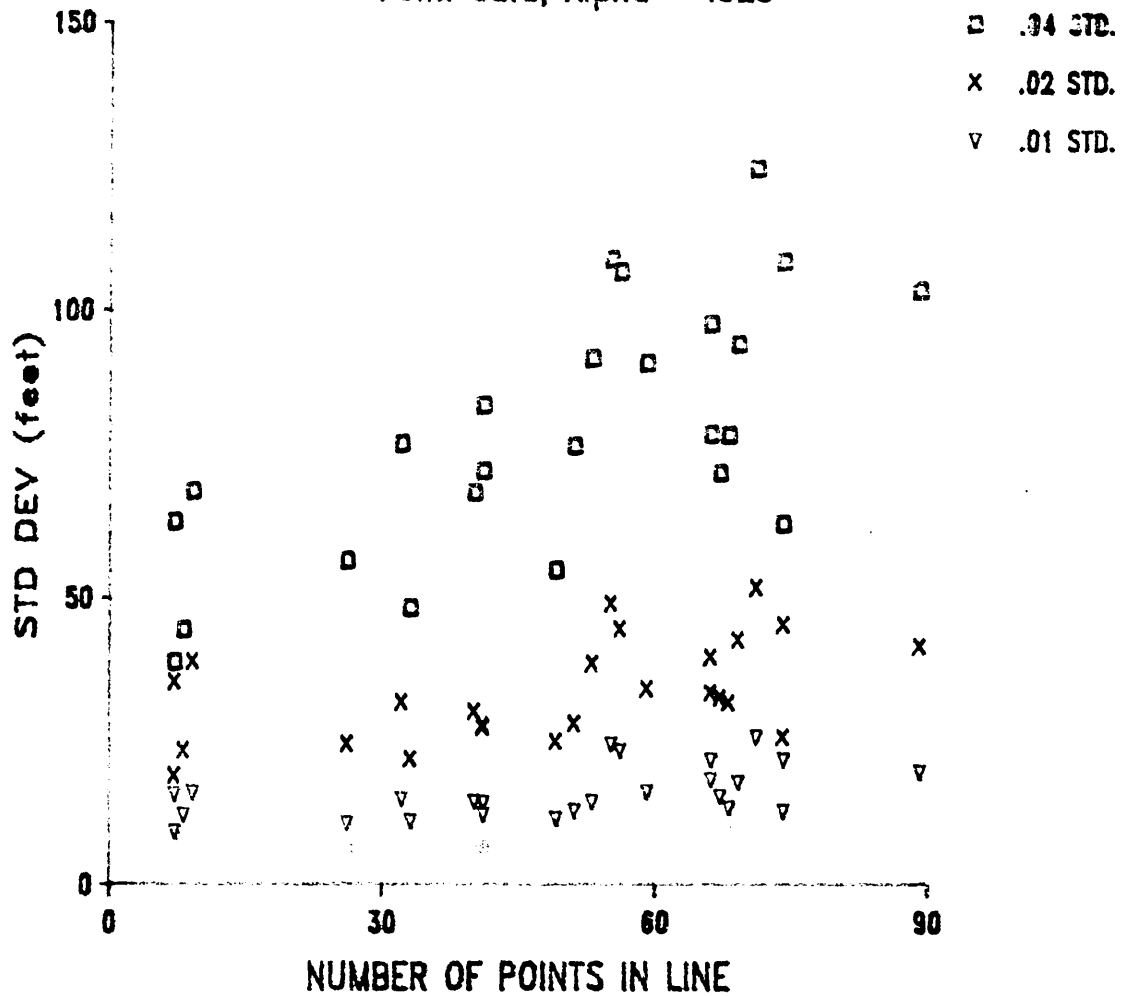


Figure 31: Standard deviation of line length plotted against number of point per line for point mode simulations at alpha .025.

SUMMARY AND CONCLUSION

The objectives of this study were to statistically model digitizing error, and to evaluate the effect of digitizing error upon polygon size and line length at varying map accuracy standards. Data was collected from several experienced digitizing operators in order to develop a model of digitizing error. From the error data collected, the author concluded that a digitizing operator made as many errors to one side of a line as to the other side. Four of the operators tested did have a slight tendency to create more errors on the left side of the line. One operator appeared to truly have no bias in digitizing error, and this operator was the most experienced and probably the most "careful" of the operators. The collected error data showed that the mean perpendicular error ranged from .001 inches in stream mode to .024 inches in point mode, with point mode digitizing being less variable.

Stream mode digitizing error was modelled using an ARMA (1,0) procedure from time series analysis. The stream mode data exhibited all of the characteristics of a time series which included serial correlation of the errors and regularly spaced data in time or distance. The model fit the data extremely well and indicated a strong autocorrelation among the stream mode errors. Analysis of the point mode data revealed two types of point mode digitizing, discrete and continuous point mode. Continuous point mode is characterized by

serial correlation between the errors, and results from a digitizing operator "simulating" stream mode when actually digitizing in point mode. Discrete point mode exhibited random error at each point uncorrelated with any previous point. This type of digitizing is only possible when a digitizing operator consciously stops and realigns the digitizing cursor as he/she moves from point to point. This study modelled discrete point mode as a random error with no attempt made to model continuous point mode. Continuous point mode can not be modelled using time series analysis because the points are not recorded at a regular time or distance interval. Point mode digitizing is highly subjective to the operators style of digitizing and is certainly deserving of further study.

Upon analysis of digitizing error when simulated at several accuracy standards, the author concluded that the effect of digitizing error on polygon area and line length is minimal, especially if the digitizing is conducted at a reasonable accuracy standard. In the author's opinion, a digitizing accuracy standard of .005 inches is almost impossible to attain, and a standard of .04 inches is an overly large standard. Accuracy standard bandwidths of .01 inches or .02 inches should be attainable especially at the .05 and .10 alpha levels. The National Map Accuracy Standard of .02 inches at .10 alpha is certainly within reason.

Essentially no bias resulted when digitizing polygons because of the compensating effect of digitizing error. The variation in polygon area increased as the accuracy standards became less stringent with a doubling of standard deviation occurring as accuracy standard bandwidth doubled. Even at the loosest accuracy standard, polygon area varied only by 1.677 percent of total area for stream mode digitizing. Bias and variance of line length both increased as the accuracy standard increased. A doubling of the accuracy standard bandwidth caused a quadrupling of the line length bias and a doubling to tripling of the line length standard deviation. For stream mode digitizing, the reasonable accuracy standards discussed above displayed a line length bias of less than 2 percent of total length and standard deviation of less than 1 percent of total length. Discrete point mode digitizing resulted in more precise

and more accurate area and length measurements. The variance of area or length in stream mode was 1.5 times that of point mode. The bias in line length in stream mode was three times larger the bias when digitizing in point mode. The bias of polygon area was zero for both point mode and stream mode.

The size of a feature in acres or feet and the number of points recorded for a feature affected bias and variance of feature size. Smaller polygons tended to have more bias and more variance relative to their size rather than larger polygons. Therefore, the potential for error is greater when digitizing small polygons. Stream mode digitizing error caused bias and standard deviation to increase with increasing line length (or number of points), but the bias and standard deviation as a percent of total line length remained constant as feature size changed. Point mode line bias and variance responded in a similar manner only when compared to the number of points per line. The error in map variables appears to be correlated with the number of points recorded when digitizing in point mode. For stream mode, the number of points and feature size produce similar responses since the two are so strongly correlated. Due to the control exerted by the operator when selecting points to digitize in point mode, changes in bias and variance are confounded by the effect of line complexity. This effect of line complexity on digitizing error presents an area for further study.

Prior to this study, the effect that manual digitizing error has on the accuracy of the map product in terms of map variables had not been well studied or quantified. As a result of this thesis, the author concluded that the effect of digitizing error is small and can be controlled by using the appropriate digitizing accuracy standards. Digitized maps become more expensive to produce and purchase as accuracy standards become more demanding, and the user should be able to judge whether the cost of increased accuracy is worthwhile. The results of this study provide a means of evaluating digitizing standards by showing how various digitizing standards effect the accuracy and precision of map variables, such as polygon area and line length. The next step would be to assign costs and benefits to the digitizing accuracy

standards and resulting errors so that a GIS user can select a digitizing standard that will result in a map which meets a desired level of accuracy.

LITERATURE CITED

- Allder, William and Atef A. Elassal. 1983. Digital line graphs from 1:24000 scale maps. USGS circular 895-C. 79 pp.
- Balci, Osman. 1985. Lecture notes: Modeling, simulation, and gaming (4150). Department of Computer Science, VPI&SU: Blacksburg, VA.
- Baugh, Ian D.H. and J.R. Boreham. 1976. Measuring the coastline from maps: a study of the scottish mainland. *Cartographic Journal*. 13:167-171.
- Bennett, H.C. 1977. The geographic data base - reliability or chaos? *Proc. American Congress on Surveying and Mapping*. pp. 675-80.
- Blakeman, David A. 1987. Some thoughts about GIS data entry. *GIS 87*. San Francisco. pp. 295-299.
- Blakemore, M. 1984. Generalisation and error in spatial databases. *Cartographica* 21:131-9.
- Box, George E. P. and Gwilym M. Jenkins. 1976. *Time series analysis forecasting and control*. Holden Day: San Francisco, CA. 575 pp.
- Boyle, A.R. 1970. The quantised line. *Cartographic Journal*. 7(2) 91-94.
- Burrough, P.A. 1986. *Principles of GIS for land resources assessment*. Oxford University Press: New York. 193 pp.
- Chrisman, Nicholas P. 1982. A theory of cartographic error and its measurement in digital data bases. *Proc. Auto-carto 5*. Crystal City, VA. pp. 159-168.
- 1983. Report 3. NCDCCDS. Columbus, OH. pp.23-31.
- 1983. The role of quality information in the long term functioning of GIS. *Proc. Auto-carto 6*. Ottawa. pp.303-312.
- 1984. Report 4. NCDCCDS. Columbus, OH. pp. 43-64.

- Conklin, J.E. 1957. Effect of control lag performance in a tracking task. *Journal of Exp. Psychology*. 53:261-68.
- Douglas, D.H. and T.K. Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Canadian Cartographer*. 10(2):112-122.
- Elassal, Atef and Vincent Caruso. 1983. Digital elevation models. USGS circular 895-B. 40 pp.
- Fegeas, Robin, G. 1978. The graphic input procedure: an operational line segment/polygon graphic to digital conversion. *Harvard papers on GIS (vol.7)*.
- Goodchild, Michael F. 1980. Fractals and the accuracy of geographical measures. *Mathematical Geology*. 12(2):85-98.
- Goodchild, Michael F. and Odette Dubuc. 1987. A model of error for choropleth maps with applications to geographic information systems. *Proc. Auto-carto 8*. Baltimore. pp.165-174.
- Gossard, Terry W. 1986. Line trace: an interactive line following, tagging, and editing system. *Proc. GIS Workshop*. Atlanta. pp.155-160.
- Honeycutt, D.M. 1986. Epsilon, generalization, and probability in spatial data bases. Unpublished research, University of Western Ontario. 25pp.
- Jenks, G.F. 1981. Lines, computers, and human frailties. *Annals of Association of American Geographers*. 71(1):1-10.
- MacDougal, E.B. 1975. The accuracy of map overlays. *Landsc.Plan.* 2:23-30.
- Mandlebrot, B.B. 1982. *The fractal geometry of nature*. Freeman: New York. 325pp.
- Marble, D.F., J.P. Lauzan, and McGranaghan. 1984. Development of a conceptual model of the manual digitizing process. *Proc. Int. symposium on spatial data handling*. Zurich. pp.146-171.
- Marsden, Lloyd E. 1960. How the national map accuracy standards were developed. *Surveying and Mapping*. 20(4):427-439.
- McCormac, Jack C. 1985. *Surveying*. Prentice-Hall, Inc: Englewood-Cliffs, NJ. pp.381.
- Merchant, Dean C. 1982. Report 1. NCDCCDS. Columbus, OH. pp. 27-36.
- 1987. Spatial accuracy specification for large scale topographic maps. *Photo.Eng. and Rem. Sens.* 53:958-961.
- Moellering, Harold. 1982. Report 1. NCDCCDS. Columbus, OH. pp. 1-16.
- Monmonier, Mark. 1982. *Computer-assisted cartography*. Prentice-Hall: Englewood Cliffs, N.J. 214 pp.
- Monserud, Robert A. 1986. Time-series analysis of tree-ring chronologies. *For.Sci.* 32:349-372.
- O'Neill, Michael P. and David M. Mark. 1987. The Psi-s plot: a useful representation for digital cartographic lines. *Proc. Auto-carto 8*. Baltimore. pp.231-240.

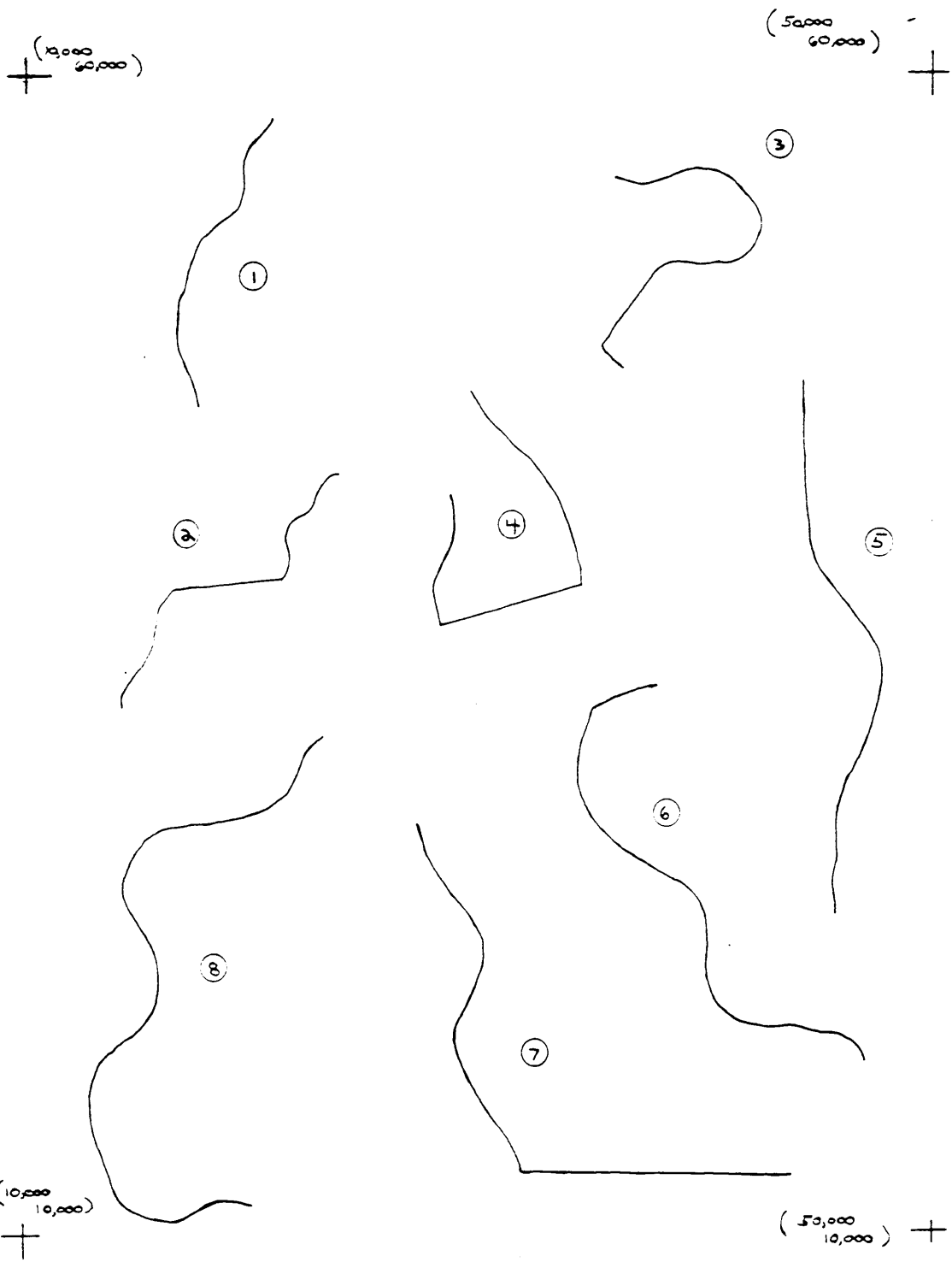
- Opheim, H. 1982. Fast reduction of a digitized curve. *Geo-Processing*. 2:33-40.
- Otawa, Toru. 1987. Accuracy of digitizing: overlooked factor in GIS operations. *Proc. GIS 87*. San Francisco. pp.295-299.
- Perkal, J. 1956. On epsilon length. *Bulletin de l'academie polonaise des sciences*. 4:399-403.
- Petersohn, C. and A.P. Vonderohe. 1982. Site specific accuracy of digitized property maps. *Proc. Auto-carto 5*. Crystal City, VA. pp.607-619.
- Peucker, T.K. 1975. A theory of the cartographic line. *Proc. Auto-Carto 2*. pp.508-518.
- 1982. Looking at computer cartography. *Geojournal*. 6:241-249.
- Poulton, E.C. 1962. On simple methods of scoring tracking errors. *Psychology Bulletin*. 59:320-328.
- Prisley, S.P. 1986. Commercial geographic information systems for natural resources management: what a manager needs to know. *Proc. GIS workshop*. Atlanta. pp.1-11.
- SAS Institute Inc. 1982. *SAS user's guide: basics*. SAS Institute Inc.: Cary, NC. 923pp.
- Shea, K. Stewart. 1987. Cartographic data entry through automatic feature tracking. *Proc. Auto-carto 8*. Baltimore. pp.660-669.
- Sheehand, D.E. 1979. A discussion of the SYMAP program. *Harvard Library of Computer Graphics mapping collection: mapping software and cartographic databases*. 2:167-179.
- Shumway, Clyde. 1986. Summary of forest service GIS activities. *Proc. GIS Workshop*. Atlanta. pp.49-52.
- Stolk, Rene, and George Ettershank. 1987. Calculating the area of an irregular shape. *Byte*. :135-136.
- Surveys and mapping branch of canadian map office. 1976. A guide to the accuracy of maps: technical report service. Canada map office, Ottawa. 12 pp.
- Thompson, M.M. 1960. A current view of the national map accuracy standards. *Surveying and Mapping*. 16(2):449-457.
- 1979. *Maps for America*. USGS:Washington,D.C. 265 pp.
- Thompson, M.M. and G.H. Rosenfield. 1971. On map accuracy specifications. *Surveying and Mapping*. 31:57-64.
- Traylor, Charles T. 1979. The evaluation of a methodology to measure manual digitizing error in cartographic data bases. Ph.D. disseration. University of Kansas. 117 pp.
- Vitek, John D. and Donald G. Richards. 1978. Incorporating inherent map error into flood-hazard analysis. *Professional Geographer*. 30(2):168-173.
- Vonderohe, Alan P. and Nicholas R. Chrisman. 1985. Tests to establish the quality of digital cartographic data: some examples from the Dane County Land Records Project. *Proc. Auto-carto 7*. Washington,D.C. pp. 552-559.

Wakeley, Richard R. 1987. GIS and Weyerhauser - 20 years experience. Proc. GIS 87. San Francisco. pp.446-455.

Walsh, Stephen J., Dale R. Lightfoot, and David R. Butler. 1987. Recognition and assessment of error in geographic information systems. Photo. Eng. and Rem. Sens. 53(10):1423-1430.

Zar, Jerrold H. 1984. *Biostatistical analysis*. Prentice-Hall Inc.: Englewood Cliffs, NJ. 718pp.

Appendix A. Test Features



(10,000
60,000)

(50,000
60,000)

9

11

12

10

13

16

14

15

(10,000
10,000)

(50,000
10,000)

Appendix B1. Error Measurement Program

```
C#####  
C#####  
C  
C          P*E*D  
C  
C          PERPENDICULAR ERROR DISTANCE CALCULATOR  
C  
C          version 3.0  
C  
C          Brent Keefer  
C          GIS/Remote Sensing Section  
C          Department of Forestry  
C          Virginia Tech  
C  
C          September 1987  
C  
C          Description: This program inputs two digitized linear features  
C          using free format and calculates the distance  
C          from a point on one line (sample line) to where  
C          there is a perpendicular intersection with the  
C          other line (standard line).  
C          Input format is an X coordinate followed by  
C          at least one space and then a Y coordinate. Both  
C          coordinates must be real numbers. Linear features  
C          are separated by a coordinate entry of 0 and  
C          a file is ended by a coordinate entry of 1.  
C  
C          This program is written in MS-FORTRAN and is designed  
C          to run on an IBM-PC AT using DOS operating system.  
C#####  
C#####
```

```

C
C      *** MAIN PROGRAM ***
C
C      @ Variable Declaration @
C
INTEGER MARK1, SIG1, FLAG, LAG, LOOP
REAL SX1, SY1, SX2, SY2, TX, TY, XINT, YINT, DIS, LAGX2, LAGY2, SIG
CHARACTER*18 IN1, IN2, OUT, OUT2
C
C      @ Initialize Variables @
C
MARK1 = 0
SIG1 = 0
FLAG = 0
SIG = 0.0
LAG = 0
LOOP = 0
C
C      @ Keyboard Display and Input @
C
WRITE(*, '(//////////)')
WRITE(*, 10)
10  FORMAT(35x, 'P * E * D '/')
WRITE(*, 15)
15  FORMAT(20x, 'Perpendicular Error Distance Calculator '///)
WRITE(*, 20)
20  FORMAT(35x, 'version 3.0/')
WRITE(*, 25)
25  FORMAT(34x, 'Brent Keefer')
WRITE(*, 30)
30  FORMAT(34x, 'Virginia Tech')
WRITE(*, 35)
35  FORMAT(34x, 'September 1988'////////)
PAUSE '          Press return to continue '
C
C
WRITE(*, '(//////////)')
WRITE(*, 50)
50  FORMAT(10x, 'Enter filename for standard feature data ')
READ(*, 55) IN1
55  FORMAT(A18)
WRITE(*, 60)
60  FORMAT(10x, 'Enter filename for sample feature data ')
READ(*, 65) IN2
65  FORMAT(A18)
WRITE(*, 70)
70  FORMAT(10x, 'Enter filename for error distance output ')
READ(*, 80) OUT
80  FORMAT(A18)
WRITE(*, 82)
82  FORMAT(10x, 'Enter filename for runs test data output ')
READ(*, 84) OUT2
84  FORMAT(A18)
C
C
C

```



```

WRITE(*,'(////)')
WRITE(*,90)
90  FORMAT(10x,'Performing calculations ... ')
C
C      @ Open Files @
C
OPEN(1,FILE = IN1,STATUS = 'OLD')
OPEN(2,FILE = IN2,STATUS = 'OLD')
OPEN(3,FILE = OUT,STATUS = 'NEW')
OPEN(4,FILE = OUT2,STATUS = 'NEW')
C
C      @ Main Program @
C      : Read coordinates from standard and sample files
C      : Call subroutine to determine perpendicular intercept
C      : Calls subroutines to determine position of intersection
C
C
100 READ(1,*) SX1,SY1
200 READ(1,*) SX2,SY2
C
250 IF(SX2.EQ.0.OR.SX2.EQ.1) GOTO 500
300 IF(MARK1.EQ.0) THEN
    READ(2,*) TX,TY
    IF(TX.EQ.0.OR.TX.EQ.1) GOTO 600
ENDIF
C
CALL INTER(SX1,SY1,SX2,SY2,TX,TY,XINT,YINT)
CALL CHECK1(SX1,SX2,XINT,SY1,SY2,YINT,SIG1)
C
C      @ Main Program Continued @
C      : If intersection is between current standard points
C      : calculate distance, else check for forward position
C
IF(SIG1.EQ.0) THEN
    CALL DIST(XINT,YINT,TX,TY,DIS,SX1,SX2,SY1,SY2)
    MARK1=0
    GOTO 300
ELSE
    CALL CHECK2(XINT,SX1,SX2,MARK1,SY1,SY2,YINT)
ENDIF
C
C      @ Main Program Continued @
C      : If intersection is ahead of current standard points
C      : advance to next pair of standard points.
C
IF(MARK1.EQ.1) THEN
    SX1=SX2
    SY1=SY2
    IF(FLAG.EQ.0) GOTO 200
    IF(FLAG.EQ.2) THEN
        SX2=LAGX2
        SY2=LAGY2
        FLAG=0
        GOTO 250
    ENDIF

```

```

ENDIF
C
C      @ Main Program Continued @
C      : If intersection is behind current standard points,
C      call subroutine behind to determine if unable to
C      construct perpendiculars or if a doubling back problem.
C
IF(MARK1.EQ.2) THEN
  CALL BEHIND(SX1,SY1,TX,TY,LAGX2,LAGY2,FLAG,SX2,SY2,LAG,LOOP)
  IF(FLAG.EQ.2) THEN
    MARK1=0
    GOTO 300
  ENDIF
  IF(FLAG.EQ.1) THEN
    SX1=SX2
    SX2=LAGX2
    SY1=SY2
    SY2=LAGY2
    FLAG=0
    GOTO 250
  ENDIF
ENDIF
ENDIF
C
C      @ Main Program Continued @
C      : Determines if end of file or new line
C
500 IF(SX2.EQ.0) THEN
  SIG = 2.0
  CALL RUNS(SIG)
  WRITE(3,*) 'NEW LINE'
520 IF(TX.EQ.0) THEN
  MARK1=0
  GOTO 100
  ELSE
  READ(2,*) TX,TY
  GOTO 520
ENDIF
ELSE
  SIG = 2.0
  CALL RUNS(SIG)
  WRITE(3,525) LAG,LOOP
525  FORMAT('END OF FILE',2X,'LAG',I5,2X,'LOOP',I5)
ENDIF
C
600 IF(TX.EQ.0) THEN
  SIG = 2.0
  CALL RUNS(SIG)
  WRITE(3,*) 'NEW LINE'
620 IF(SX2.EQ.0) THEN
  MARK1=0
  GOTO 100
  ELSE
  READ(1,*) SX2,SY2
  GOTO 620
ENDIF
ELSE

```

```

        SIG = 2.0
        CALL RUNS(SIG)
        WRITE(3,625) LAG,LOOP
625    FORMAT('END OF FILE',2X,'LAG',I5,2X,'LOOP',I5)
    ENDIF
999    STOP
    END
C
C#####
C
C
C    SUBROUTINE INTER - Calculates intersection of line from sample
C    point to a perpendicular intersection with standard line.
C
C    SUBROUTINE INTER(SX1,SY1,SX2,SY2,TX,TY,XINT,YINT)
C    REAL SX1,SY1,SX2,SY2,TX,TY,XINT,YINT,A,B,C,D,E,F,DENOM
C
C    A=SY1-SY2
C    B=SX2-SX1
C    E=(SY1*SX2)-(SY2*SX1)
C    C=SX2-SX1
C    D=SY2-SY1
C    F=(SY2-SY1)*TY+(SX2-SX1)*TX
C    DENOM=A*D-B*C
C    XINT=(E*D-F*B)/DENOM
C    YINT=(A*F-C*E)/DENOM
C    RETURN
C    END
C
C#####
C
C    SUBROUTINE CHECK1 - Checks to see if intersection is between
C    current standard points.
C
C    SUBROUTINE CHECK1(SX1,SX2,XINT,SY1,SY2,YINT,SIG1)
C    REAL SX1,SX2,XINT,Q,V,SY1,SY2,YINT,W,O
C    INTEGER SIG1
C
C    Q=ABS(SX2-XINT)+ABS(SX1-XINT)
C    V=ABS(SX2-SX1)
C    W=ABS(SY2-YINT)+ABS(SY1-YINT)
C    O=ABS(SY2-SY1)
C    IF(SX1.EQ.SX2) THEN
C        IF(W.LE.O) THEN
C            SIG1=0
C            GOTO 9010
C        ELSE
C            SIG1=1
C            GOTO 9010
C        ENDIF
C    ENDIF
C    IF(Q.LE.V) THEN
C        SIG1=0
C    ELSE
C        SIG1=1

```

```

ENDIF
9010 RETURN
END
C
C#####
C
C    SUBROUTINE DIST - Calculates perpendicular error distance
C    and writes distance to output file.
C
C    SUBROUTINE DIST(XINT,YINT,TX,TY,DIS,SX1,SX2,SY1,SY2)
C    REAL XINT,YINT,TX,TY,DIS,SX1,SX2,SY1,SY1,SIG
C
C    CALL ESIGN(TX,TY,SX1,SX2,SY1,SY2,SIG)
C    DIS = SQRT((XINT-TX)**2 + (YINT-TY)**2)
C    DIS = DIS*SIG
C    WRITE(3,9600) DIS
9600 FORMAT(F8.3)
RETURN
END
C
C#####
C
C    SUBROUTINE CHECK2 - Checks to see if intersection is ahead of
C    current standard points.
C
C    SUBROUTINE CHECK2(XINT,SX1,SX2,MARK1,SY1,SY2,YINT)
C    REAL XINT,SX1,SX2,SY1,SY2,YINT
C    INTEGER MARK1
C
C    IF(SX1.EQ.SX2) THEN
C    IF(((SY2.GT.SY1).AND.(YINT.GT.SY2)).OR.((SY2.LT.SY1).AND.
C    $ (YINT.LT.SY2))) THEN
C    MARK1 = 1
C    GOTO 9650
C    ELSE
C    MARK1 = 2
C    GOTO 9650
C    ENDIF
C    ENDIF
C    IF(((SX2.GT.SX1).AND.(XINT.GT.SX2)).OR.((SX2.LT.SX1).AND.
C    $(XINT.LT.SX2))) THEN
C    MARK1 = 1
C    ELSE
C    MARK1 = 2
C    ENDIF
9650 RETURN
END
C
C#####
C
C    SUBROUTINE BEHIND - If intersection is in position
C    where perpendiculars can not be constructed, error
C    distance is omitted, and file advanced to next sample
C    point.
C    If the feature is looping back on itself, current
C    standard coordinates are adjusted properly.

```

```

C
C
SUBROUTINE BEHIND(SX1,SY1,TX,TY,LAGX2,LAGY2,FLAG,SX2,SY2,LAG,LOOP)
REAL SX1,SY1,TX,TY,FPDIS,SX2,SY2,XINT,YINT,NDIS,A,B,C,D,E,F,DENOM
REAL LAGX1,LAGX2,LAGY1,LAGY2,SIG,S1DIS
INTEGER FLAG,SIG1,MARK1,LAG,LOOP
C
C     *** Move to next pair of standard points.
C
LAGX1 = SX2
LAGY1 = SY2
READ(1,*)LAGX2,LAGY2
C
C     *** Call subroutines to calculate intersection and position of
C     intersection
C
CALL INTER(LAGX1,LAGY1,LAGX2,LAGY2,TX,TY,XINT,YINT)
CALL CHECK1(LAGX1,LAGX2,XINT,LAGY1,LAGY2,YINT,SIG1)
IF(SIG1.EQ.0) GOTO 900
CALL CHECK2(XINT,LAGX1,LAGX2,MARK1,LAGY1,LAGY2,YINT)
IF(MARK1.EQ.1) GOTO 900
C
C     *** Perpendiculars can not be constructed to sample point
C     or sample point is a lagging first point.
C     Omit sample point and advance to next point.
C
FLAG = 2
LAG = LAG + 1
GOTO 999
C
C     *** Problem of line sharply looping back itself. Advance
C     standard points to next pair of standard points
C
900 FLAG = 1
LOOP = LOOP + 1
C
999 RETURN
END
C
C#####
C
C     SUBROUTINE ESIGN - Calculates the sign of the error.
C
SUBROUTINE ESIGN(TX,TY,SX1,SX2,SY1,SY2,SIG)
REAL TX,TY,SX1,SX2,SY1,SY2,ASY1,ANG,SIG,ASX1
C
SIG = 1.0
ASY1 = SY1
ASX1 = SX1
C
IF(TY-SY1.EQ.0.0.OR.SY2-SY1.EQ.0.0) THEN
    ASY1 = SY1 + 0.01
ENDIF
IF(TX-SX1.EQ.0.0.OR.SX2-SX1.EQ.0.0) THEN
    ASX1 = SX1 + 0.01
ENDIF

```

```

ANG = ATAN2(TY-ASY1, TX-ASX1) - ATAN2(SY2-ASY1, SX2-ASX1)
ANG = SIGN(3.1415926, ANG) - ANG
SIG = SIGN(SIG, ANG)
CALL RUNS(SIG)
RETURN
END
C
C#####
C
C  SUBROUTINE RUNS - Calculates number of negative points, number
C    positive points, and number of runs for each line segment.
C    Outputs data to file.
C
C  SUBROUTINE RUNS(SIG)
C    REAL SIG
C    INTEGER PRIOR, NPOS, NNEG, NRUNS
C
C    DATA PRIOR, NPOS, NNEG, NRUNS/4*0.0/
C
C    IF(SIG.EQ.2.0) GOTO 100
C    IF(SIG.EQ.1.0) NPOS = NPOS + 1
C    IF(SIG.EQ.-1.0) NNEG = NNEG + 1
C
C    IF(PRIOR.EQ.0) GOTO 90
C    IF(PRIOR.LT.INT(SIG).OR.PRIOR.GT.INT(SIG)) NRUNS = NRUNS + 1
C

```

```
90  PRIOR = INT(SIG)
    GOTO 900
C
C
100 WRITE(4,*) NPOS,NNEG,NRUNS
    NPOS = 0
    NNEG = 0
    NRUNS = 0
    PRIOR = 0
C
900 RETURN
    END
```

Appendix B2. SIMDIGER

```

C*****
C*****
C**          SIMDIGER          **
C**                                **
C**          SIMULATED DIGITIZING ERROR          **
C**                                **
C** Purpose: This program uses an ARMA(1,0) model of **
C** stream mode digitizing error to simulate polygons **
C** and lines on a point by point basis. Point mode **
C** digitizing is simulated using a uniform random model. **
C**                                **
C**                                **
C**                                **
C**          Brent Keefer          **
C**          GIS/Remote Sensing Section          **
C**          Department of Forestry          **
C**          Virginia Tech          **
C**                                **
C**                                **
C**          Version 3.0          **
C**          August 1988          **
C**                                **
C** Description: This program is written in **
C** VAX FORTRAN and is designed to run on a MICRO - **
C** VAX II using VMS operating system.          **
C**                                **
C*****
C*****
C
C          VARIABLE DECLARATION
C
C          REAL*8 WIDTH,E2,LEN(500),ALPHA(4),STANDARD(4),LEVEL(4)
C          REAL*8 SX1,SY1,SX2,SY2,SCORX(500),SCORY(500),OUT(500),SUMERR
C          REAL*8 XMN,STD,DIR,EDIS,EBAR(500),ERRMAX,ECORY(500),ECORX(500)
C          INTEGER FLAG,N,COUNT,I,NUMB
C          INTEGER*4 ISEED
C          CHARACTER*15 ERROR,ERRCOR,TYPE,ANS,MODE
C          CHARACTER*20 INCOR,OUTPUT,INFILE,OUTFILE,ABSUM,RELSUM
C
C          DATA ALPHA/.4464285,.5102041,.6079028,.78125/
C          DATA LEVEL/.005,.01,.02,.04/

```



```

DATA STANDARD/10.0,20.0,40.0,80.0/
C
C **** KEYBOARD INPUT ****
C
DO 102 I = 1,50
WRITE(*,*) ''
102 CONTINUE
WRITE(*,*) ' *****'
WRITE(*,*) ' *****'
WRITE(*,*) ''
WRITE(*,*) ' WELCOME TO SIMDIGER '
WRITE(*,*) ''
WRITE(*,*) ' DEVELOPED BY '
WRITE(*,*) ' BRENT KEEFER '
WRITE(*,*) ' VPI&SU FORESTRY DEPT.'
WRITE(*,*) ''
WRITE(*,*) ' AUG. 1988'
WRITE(*,*) ' VERSION 3.0'
WRITE(*,*) ''
WRITE(*,*) ' *****'
WRITE(*,*) ' *****'
C
WRITE(*,*) ''
WRITE(*,*) ''
103 WRITE(*,104)
104 FORMAT(10X,' Enter the seed value for random number gen. ')
READ(*,*,ERR = 103) ISEED
C
C
C
100 WRITE(*, '(//)')
WRITE(*,*) ' Enter filename containing source polygon or line
$ filenames '
READ(*,150) INFILE
150 FORMAT(A20)
WRITE(*,*) ''
WRITE(*,*) ' Enter filename containing output filenames '
READ(*,152) OUTFILE
152 FORMAT(A20)
WRITE(*,*) ''
WRITE(*,*) ' Enter filename for absolute summary output '
READ(*,152) ABSUM
WRITE(*,*) ''
WRITE(*,*) ' Enter filename for relative summary output '
READ(*,152) RELSUM
WRITE(*,*) ''
WRITE(*,*) ' Select simulation type: (P)olygon,(L)ine,(LO)cations'
READ(*,154) TYPE
154 FORMAT(A15)
WRITE(*,*) ''
WRITE(*,*) ' Select: (S)treem or (P)oint mode'
READ(*,156) MODE
156 FORMAT(A15)
WRITE(*,*) ''
WRITE(*,*) ' Enter number of desired simulations '
READ(*,*) N

```

```

WRITE(*,*) ' '
WRITE(*,*) ' Enter scale factor '
READ(*,*) SCALE
C
C **** OPEN TEXT CONTROL FILES ****
C
OPEN(3,FILE=INFILE,STATUS='OLD')
OPEN(4,FILE=OUTFILE,STATUS='OLD')
OPEN(8,FILE=ABSUM,STATUS='NEW')
OPEN(9,FILE=RELSUM,STATUS='NEW')
C
C **** READ INPUT AND OUTPUT FILES FOR PROCESSING ****
C
180 READ(3,181,END=999) INCOR
READ(4,181,END=999) OUTPUT
181 FORMAT(A20)
WRITE(*,*) ' '
WRITE(*,183) INCOR
183 FORMAT(5X,'Processing file ',A20)
C
C **** OPEN DATA FILES TO BE PROCESSED ****
C
OPEN(1,FILE=INCOR,STATUS='OLD')
OPEN(2,FILE=OUTPUT,STATUS='NEW')
C
C **** MAIN DRIVER PROGRAM ****
C The exterior loop (k) controls the alpha level of the accuracy
C standard, and the interior loop (L) controls the band width
C of the accuracy standard. Every input feature to be simulated
C is simulated 16 times using the combinations of K and L.
C
DO 625 K = 1,4
WRITE(*,*) ' '
DO 600 L = 1,4
WRITE(*,192) LEVEL(K),STANDARD(L)
192 FORMAT(20X,'Alpha level 'F7.5,5X,'Accuracy standard ',F6.3)
FLAG = 0
COUNT = 1
NUMB = 1
ERRMAX = 0.0
SUMERR = 0.0
WIDTH = STANDARD(L)
ISEEDER = ISEED
C
C **** Calls point simulator if specified and calculates the actual
C area or length of the source features.
C
IF(TYPE.EQ.'LO') THEN
CALL POINTS(N,ISEED,OUT,E2)
GOTO 1000
ENDIF
IF(TYPE.EQ.'P') CALL SACALC(SCORX,SCORY,OUT,COUNT)
IF(TYPE.EQ.'L') CALL SDCALC(SCORX,SCORY,OUT,COUNT)
C
C **** Looping for specified number of simulations. Within loop

```

```

C   an array of errors are generated (EGEN), error influenced
C   coordinates are created (CORDS or PCORDS), and the area
C   or length of the error features are calculated (ACALC and
C   DCALC). Output routine (PRINT) is also called to output
C   to file the results of the simulation.
C
DO 900 I = 1,N
  E2 = (ALPHA(K)*WIDTH)**2
  IF(MODE.EQ.'S') CALL EGEN(LEN,E2,ISEED)
  SX1 = SCORX(1)
  SY1 = SCORY(1)
200  SX2 = SCORX(NUMB + 1)
  SY2 = SCORY(NUMB + 1)
  IF(SX2.LT.0.AND.TYPE.EQ.'P') THEN
    CALL ACALC(N,FLAG,I,COUNT,OUT,ECORX,ECORY)
  ENDIF
  IF(SX2.LT.0.AND.TYPE.EQ.'L') THEN
    CALL DCALC(N,FLAG,I,COUNT,OUT,ECORX,ECORY)
  ENDIF
  IF(FLAG.EQ.1) GOTO 800
  SUMERR = SUMERR + ABS(LEN(NUMB))
  IF(ABS(LEN(NUMB)).GT.ERRMAX) ERRMAX = ABS(LEN(NUMB))
  IF(MODE.EQ.'P') CALL PCORDS(ISEED,SX1,SY1,ECORX,
$   ECORY,NUMB,E2)
  IF(MODE.EQ.'S') CALL CORDS(SX1,SY1,SX2,SY2,LEN,ECORX,
$   ECORY,NUMB)
  SX1 = SX2
  SY1 = SY2
  GOTO 200
800  EBAR(I) = SUMERR/NUMB
  NUMB = 1
  SUMERR = 0.0
  FLAG = 0
900  CONTINUE
1000 CALL PRINT(N,OUT,ERRMAX,EBAR,INCOR,WIDTH,K,SCALE,ISEEDER,
$   MODE)
  REWIND(1)
600  CONTINUE
625  CONTINUE
C
C   CLOSE(1)
C   CLOSE(2)
C   GOTO 180
C
C
C
999  WRITE(*,*) ' '
  WRITE(*,*) '          PROGRAM IS FINISHED '
  WRITE(*,*) ' '
  STOP
  END
C
C#####
C

```

```

C  SUBROUTINE ACALC - Calculates area of error polygon.
C
C  SUBROUTINE ACALC(N,FLAG,I,COUNT,OUT,ECORX,ECORY)
C
C  REAL*8 X(500),Y(500),A,OUT(500),ECORX(500),ECORY(500)
C  INTEGER N,FLAG,I,COUNT,K
C
C  **** CALCULATE THE AREA OF AN IRREGULAR SHAPE ****
C
C  DO 2000 K=2 ,(COUNT + 1)
C    X(K) = ECORX(K-1)
C    Y(K) = ECORY(K-1)
2000  CONTINUE
C  X(1) = X(COUNT + 1)
C  Y(1) = Y(COUNT + 1)
C  A=0.0
C  DO 2500 K=1, COUNT
C    A = A + X(K)*Y(K + 1) - X(K + 1)*Y(K)
2500  CONTINUE
C  A = .5 * ABS(A)
C
C  **** WRITE AREA TO ARRAY ****
C
C  OUT(I + 1) = A
C  FLAG = 1
C  RETURN
C  END
C
C#####
C
C  SUBROUTINE DCALC - Calculates the length of error line.
C
C  SUBROUTINE DCALC(N,FLAG,I,COUNT,OUT,ECORX,ECORY)
C
C  REAL*8 X(500),Y(500),LENGTH,OUT(500),ECORX(500),ECORY(500)
C  INTEGER N,FLAG,I,COUNT,K
C
C  **** CALCULATE THE LENGTH OF A LINE ****
C
C  LENGTH = 0
C  DO 6500 K=1,(COUNT-1)
C    LENGTH=LENGTH + SQRT((ECORX(K)-ECORX(K + 1))**2 +
C  # (ECORY(K)-ECORY(K + 1))**2)
6500  CONTINUE
C
C  **** WRITE LENGTH TO ARRAY ****
C
C  OUT(I + 1) = LENGTH
C  FLAG = 1
C  RETURN
C  END
C
C#####
C

```

```

C     SUBROUTINE CORDS - Calculates the error coordinates for
C     a stream-mode simulation.
C
C     SUBROUTINE CORDS(SX1,SY1,SX2,SY2,LEN,ECORX,ECORY,NUMB)
C
C     REAL*8 SX1,SY1,SX2,SY2,EDIS,TX,TY,TTX,TTY,ECORX(500)
C     REAL*8 BEAR,EBEAR,ECORY(500),LEN(500)
C     INTEGER COUNT,NUMB
C
C     IF(SX2-SX1.EQ.0.0) SX2 = SX2 + .0001
C     IF(SY2-SY1.EQ.0.0) SY2 = SY2 + .0001
C
C     **** CALCULATE BEARING OF STANDARD LINE SEGMENT ****
C
C     BEAR = ATAN2(SX2-SX1,SY2-SY1)*180.0/3.1415926
C     IF(BEAR.GE.0.AND.SX2-SX1.GT.0) BEAR = BEAR
C     IF(BEAR.GE.0.AND.SX2-SX1.LT.0) BEAR = ABS(BEAR) + 180.0
C     IF(BEAR.LT.0.AND.SX2-SX1.GT.0) BEAR = BEAR + 180.0
C     IF(BEAR.LT.0.AND.SX2-SX1.LT.0) BEAR = BEAR + 360.0
C
C     **** CALCULATE BEARING OF PERPENDICULAR ERROR DISTANCE ****
C
C     EDIS = LEN(NUMB)
C     IF(EDIS.GE.0) EBEAR = BEAR-90.0
C     IF(EDIS.LT.0) EBEAR = BEAR + 90.0
C     IF(EBEAR.LT.0) EBEAR = EBEAR + 360.0
C     IF(EBEAR.GT.360) EBEAR = EBEAR-360.0
C
C     **** CALCULATE COORDINATES OF ERROR POINT AND WRITE TO FILE ***
C
C     TTX = SIN(EBEAR*3.1415926/180.0)*ABS(EDIS)
C     TTY = COS(EBEAR*3.1415926/180.0)*ABS(EDIS)
C     TX = TTX + SX1
C     TY = TTY + SY1
C     ECORX(NUMB) = TX
C     ECORY(NUMB) = TY
C     NUMB = NUMB + 1
C     RETURN
C     END
C
C #####
C
C     SUBROUTINE EGEN - Generates error distance for stream-mode
C     simulation by using ARMA(1,0) model.
C
C     SUBROUTINE EGEN(LEN,E2,ISEED)
C
C     REAL*8 LEN(500),E2,P,V2,XMN,STD,RNUM
C     INTEGER H
C     INTEGER*4 ISEED
C
C     **** CALCULATES THE CORRELATION COEFFICIENT ****
C
C     3000 RNUM = RAND(ISEED)
C     IF(RNUM.LE..0375) P = .369
C     IF(RNUM.GT..0376.AND.RNUM.LE..1626) P = .469

```

```

IF(RNUM.GT..1627.AND.RNUM.LE..2752) P = .569
IF(RNUM.GT..2753.AND.RNUM.LE..5253) P = .669
IF(RNUM.GT..5254.AND.RNUM.LE..8254) P = .769
IF(RNUM.GT..8255) P = .869
C
IF(P.LT.0.OR.P.GT.1) GOTO 3000
V2 = E2*(1-P**2)
C
C **** CALCULATES THE ERROR DISTANCES FOR THE SERIES ****
C
LEN(1) = SQRT(V2) * RNORM(0.0,1.0,ISEED) / SQRT(1-P**2)
DO 3100 H = 2,500
LEN(H) = LEN(H-1) * P + (SQRT(V2) * RNORM(0.0,1.0,ISEED))
3100 CONTINUE
RETURN
END
C
C #####
C
C FUNCTION RNORM - Normal random variate generator.
C
REAL*8 FUNCTION RNORM(XMN,STD,ISEED)
REAL*8 XMN,STD,ENORM
INTEGER*4 ISEED,IEVEN
DATA IEVEN/0/
C
C **** NORMAL RANDOM NUMBER GENERATOR, CALLS RANDOM NUMBER GENERATOR ***
C
C
IF(IEVEN.GT.1) GOTO 5100
5000 UA = 2. *RAND(ISEED) - 1
UB = 2. *RAND(ISEED) - 1
W = UA * UA + UB * UB
IF (W.GT.1.0) GOTO 5000
W = SQRT(-2. * ALOG(W)/W)
RNORM = UA*W
ENORM = UB*W
IEVEN = 2
GOTO 5200
5100 IEVEN = 1
RNORM = ENORM
5200 RNORM = RNORM*STD + XMN
RETURN
END
C
C #####
C
C FUNCTION RAND - MULTIPLICATIVE CONGRUENTIAL RANDOM
C VARIATE GENERATOR
C
REAL*8 FUNCTION RAND(ISEED)
DOUBLE PRECISION Z,D2P31M
DOUBLE PRECISION DMOD
DATA D2P31M/214783647.D0/
C
Z = ISEED

```

```

Z = DMOD(16807.D0 * Z,D2P31M)
ISEED = Z
RAND = Z/D2P31M
RETURN
END
C
C #####
C
C SUBROUTINE SACALC - Calculates the area of the source polygon.
C
C SUBROUTINE SACALC(SCORX,SCORY,OUT,COUNT)
C
C REAL*8 SCORX(500),SCORY(500),X(500),Y(500),OUT(500)
C INTEGER COUNT,T,K
C
C *** READ SOURCE POLYGON FILE INTO ARRAY SCORX AND SCORY ***
C
C T = 1
7000 READ(1,*,END = 7010) SCORX(T),SCORY(T)
C T = T + 1
C GOTO 7000
7010 COUNT = T - 3
C
C *** CALCULATE AREA OF SOURCE POLYGON ***
C
C DO 7020 K = 2,(COUNT + 1)
C X(K) = SCORX(K-1)
C Y(K) = SCORY(K-1)
7020 CONTINUE
C X(1) = X(COUNT + 1)
C Y(1) = Y(COUNT + 1)
C A = 0.0
C DO 7030 K = 1,COUNT
C A = A + X(K)*Y(K + 1) - X(K + 1)*Y(K)
7030 CONTINUE
C A = .5*ABS(A)
C
C *** WRITE AREA TO OUTPUT ARRAY ***
C
C OUT(1) = A
C RETURN
C END
C
C #####
C
C SUBROUTINE SDCALC - Calculates length of source line.
C
C SUBROUTINE SDCALC(SCORX,SCORY,OUT,COUNT)
C
C REAL*8 SCORX(500),SCORY(500),OUT(500),LENGTH
C INTEGER T,K,COUNT
C
C *** READ SOURCE LINE FILE INTO SCORX AND SCORY ARRAYS ***
C
C T = 1
7050 READ(1,*,END = 7060) SCORX(T),SCORY(T)

```

```

T = T + 1
GOTO 7050
7060 COUNT = T-3
C
C *** CALCULATE LENGTH OF LINE ***
C
LENGTH = 0.0
DO 7070 K = 1, (COUNT-1)
LENGTH = LENGTH + SQRT((SCORX(K)-SCORX(K+1))**2 +
# (SCORY(K)-SCORY(K+1))**2)
7070 CONTINUE
C
C *** OUTPUT LENGTH TO ARRAY ***
C
OUT(1) = LENGTH
RETURN
END
C
C #####
C
C SUBROUTINE POINTS - Error simulation for two locations.
C
SUBROUTINE POINTS(N, ISEED, OUT, E2)
C
REAL*8 WIDTH, X(2), Y(2), RLINE, OUT(500), EX(2), EY(2)
REAL*8 ELINE, DIR, EDIS, XMN, STD, E, E2
INTEGER*4 ISEED
INTEGER N, I, J
C
C *** READ SOURCE POINTS ***
C
8000 READ(1, *) X(1), Y(1)
READ(1, *) X(2), Y(2)
RLINE = SQRT((X(1)-X(2))**2 + (Y(1)-Y(2))**2)
OUT(1) = RLINE
C
C *** CALCULATING ERROR POINTS ***
C
E = SQRT(E2)
DO 8010 I = 1, N
WRITE(*, *) ' CREATING SIMULATION', I
DO 8020 J = 1, 2
DIR = ABS(RAND(ISEED)*360.00)
EDIS = ABS(RNORM(0.0, E, ISEED))
IF(DIR.GT.360) DIR = DIR-360
IF(DIR.LE.90) THEN
EX(J) = X(J) + SIN(DIR*3.1415926/180)*EDIS
EY(J) = Y(J) + COS(DIR*3.1415926/180)*EDIS
ENDIF
IF(DIR.LE.180.AND.DIR.GT.90) THEN
DIR = 180-DIR
EX(J) = X(J) + SIN(DIR*3.1415926/180)*EDIS
EY(J) = Y(J) - COS(DIR*3.1415926/180)*EDIS
ENDIF
IF(DIR.LE.270.AND.DIR.GT.180) THEN
DIR = DIR-180

```



```

      EX(J) = X(J) - SIN(DIR*3.1415926/180)*EDIS
      EY(J) = Y(J) - COS(DIR*3.1415926/180)*EDIS
    ENDIF
    IF(DIR.LE.360.AND.DIR.GT.270) THEN
      DIR = 360-DIR
      EX(J) = X(J) - SIN(DIR*3.1415926/180)*EDIS
      EY(J) = Y(J) + COS(DIR*3.1415926/180)*EDIS
    ENDIF
8020   CONTINUE
      ELINE = SQRT((EX(1)-EX(2))**2 + (EY(1)-EY(2))**2)
      OUT(I+1) = ELINE
8010   CONTINUE
      RETURN
      END
C
C
C #####
C
C   SUBROUTINE PCORDS - Point mode error generation.
C
C   SUBROUTINE PCORDS(ISEED,SX1,SY1,ECORX,ECORY,NUMB,E2)
C
C   REAL*8 WIDTH,X(2),Y(2),RLINE,OUT(500),EX(2),EY(2),SX1,SX2
C   REAL*8 ELINE,DIR,EDIS,ECORX(500),ECORY(500),XMN,STD,E,E2
C   INTEGER*4 ISEED
C   INTEGER N,I,J,NUMB
C
C   *** CALCULATING ERROR POINTS ***
C
      E = SQRT(E2)
      DIR = ABS(RAND(ISEED)*360.00)
      EDIS = ABS(RNORM(0.0,E,ISEED))
      IF(DIR.GT.360) DIR = DIR-360
      IF(DIR.LE.90) THEN
        ECORX(NUMB) = SX1 + SIN(DIR*3.1415926/180)*EDIS
        ECORY(NUMB) = SY1 + COS(DIR*3.1415926/180)*EDIS
      ENDIF
      IF(DIR.LE.180.AND.DIR.GT.90) THEN
        DIR = 180-DIR
        ECORX(NUMB) = SX1 + SIN(DIR*3.1415926/180)*EDIS
        ECORY(NUMB) = SY1 - COS(DIR*3.1415926/180)*EDIS
      ENDIF
      IF(DIR.LE.270.AND.DIR.GT.180) THEN
        DIR = DIR-180
        ECORX(NUMB) = SX1 - SIN(DIR*3.1415926/180)*EDIS
        ECORY(NUMB) = SY1 - COS(DIR*3.1415926/180)*EDIS
      ENDIF
      IF(DIR.LE.360.AND.DIR.GT.270) THEN
        DIR = 360-DIR
        ECORX(NUMB) = SX1 - SIN(DIR*3.1415926/180)*EDIS
        ECORY(NUMB) = SY1 + COS(DIR*3.1415926/180)*EDIS
      ENDIF
      NUMB = NUMB + 1
      RETURN
      END
C

```

```

C #####
C
C SUBROUTINE PRINT - Produces output files. One file for
C a stats summary of the N simulations,
C one file of absolute bias and absolute
C std.dev., and one file of relative bias
C and relative std. dev. summary.
C
C SUBROUTINE PRINT(N,OUT,ERRMAX,EBAR,INCOR,WIDTH,K,SCALE,ISEEDER,
C $MODE)
C
C REAL*8 OUT(500),SUM,SQDIFA,SQDIFV,SUMERR
C REAL*8 ERRMAX,EBAR(500),SCALE,MEANOUT,CALPHA(4)
C REAL*8 MSE,VAR,BIAS,MEANERR,STD
C INTEGER N,J,P,Y,T,K,KCHECK
C CHARACTER*15 STYLE,MODE
C CHARACTER*20 INCOR,OUTIT
C
C DATA CALPHA/.025,.05,.10,.20/
C DATA KCHECK/-1/
C
C **** INITIALIZE VARIABLES ***
C
C SUM = 0.0
C SQDIFA = 0.0
C SQDIFV = 0.0
C SUMERR = 0.0
C BAND = WIDTH/2000
C
C *** SCALE SIMULATED LENGTHS OR AREAS ***
C
C DO 100 L = 1,N+1
C OUT(L) = OUT(L) / SCALE
100 CONTINUE
C
C **** CALCULATE SUMS AND MEANS ***
C
C DO 9100 J = 2,(N+1)
C SUM=SUM+OUT(J)
9100 CONTINUE
C
C MEANOUT = SUM / N
C ABMEAN = MEANOUT/OUT(1)
C DO 9200 I = 2,(N+1)
C SQDIFA = SQDIFA + (OUT(I) - OUT(1))**2
C SQDIFV = SQDIFV + (OUT(I) - MEANOUT)**2
9200 CONTINUE
C
C *** COMPUTE VARIANCE AND MEAN SQUARED ERROR ***
C
C MSE = SQDIFA / N
C RMSE = SQRT(MSE)
C VAR = SQDIFV / N
C STD = SQRT(VAR)
C BIAS = SQRT(MSE-VAR)
C ABMSE = MSE/(OUT(1)**2)

```

```

ABRMSE = RMSE/OUT(1)
ABVAR = VAR/(OUT(1)**2)
ABSTD = STD/OUT(1)
ABBIAS = BIAS/OUT(1)
C
C *** COMPUTE MEAN ERROR DISTANCE
C
DO 9300 J = 1,N
SUMERR = SUMERR + EBAR(J)
9300 CONTINUE
MEANERR = SUMERR/N
ABAVERR = MEANERR/OUT(1)
ABERRMAX = ERRMAX/OUT(1)
C
C
C **** OUTPUT TO FILE ****
C
WRITE(2,850)
850 FORMAT('1',//10X,' SIMDIGER'//)
WRITE(2,802) INCOR,N
802 FORMAT(15X,'FILENAME = ',A20,2X,'SIMULATIONS ',I3//)
WRITE(2,852) ISEEDER,MODE
852 FORMAT(15X,'SEED VALUE = ',I14,5X,'MODE = ',A15/)
WRITE(2,854) BAND
854 FORMAT(15X,'ACCURACY STD. = ',F6.4/)
WRITE(2,705) CALPHA(K)
705 FORMAT(15X,'ALPHA LEVEL = ',F6.4////)
WRITE(2,706) OUT(1)
706 FORMAT(10X,' ACTUAL AREA/LENGTH = ',F14.6//)
WRITE(2,801) MEANOUT,ABMEAN
801 FORMAT(10X,' MEAN SIMULATION AREA/DIST = ',F14.6,F14.6///)
WRITE(2,800) MSE,ABMSE
800 FORMAT(10X,' MEAN SQUARED ERROR = ',F14.6,F14.6/)
WRITE(2,840) RMSE,ABRMSE
840 FORMAT(10X,' ROOT M.S.E. = ',F14.6,F14.6/)
WRITE(2,805) VAR,ABVAR
805 FORMAT(10X,' VARIANCE = ',F14.6,F14.6/)
WRITE(2,807) STD,ABSTD
807 FORMAT(10X,' STANDARD DEVIATION = ',F14.6,F14.6/)
WRITE(2,810) BIAS,ABBIAS
810 FORMAT(10X,' BIAS = ',F14.6,F14.6///)
WRITE(2,815) ERRMAX,ABERRMAX
815 FORMAT(10X,' MAXIMUM ERROR DISTANCE = ',F14.6,F14.6/)
WRITE(2,820) MEANERR,ABAVERR
820 FORMAT(10X,' AVERAGE ERROR DISTANCE = ',F14.6,F14.6/)
C
C **** OUTPUT TO ABBREVIATED FILE ***
C
IF(KCHECK.NE.K) THEN
WRITE(8,*) ''
WRITE(9,*) ''
900 WRITE(8,902) CALPHA(K),INCOR,MODE
WRITE(9,902) CALPHA(K),INCOR,MODE
902 FORMAT(2X,'Alpha level = ',F6.4,5X,A20,5X,'MODE = ',A15)
KCHECK = K
ENDIF

```

```
WRITE(8,905) BAND,OUT(1),MEANOUT,BIAS,STD  
WRITE(9,906) BAND,OUT(1),MEANOUT,ABBIAS,ABSTD  
905  FORMAT(F6.4,4F14.6)  
906  FORMAT(F6.4,4F14.6)  
C  
9900 RETURN  
END
```

Appendix C. Empirical Data

Stream Mode Digitizing

Test Feature	operator 1		operator 2		operator 3		operator 4		operator 5	
	Mean .001 in	Std.Dev. .001 in	Mean .001 in	Std.Dev. .001 in	Mean .001 in	Std.Dev. .001 in	Mean .001 in	Std.Dev. .001 in	Mean .001 in	Std.Dev. .001 in
1	-23.8	39.8	-18.4	30.1	2.9	52.1	-8.8	50.4	2.1	45.8
2	-3.4	30.3	-20.0	32.5	27.2	31.4	24.8	40.4	14.6	29.9
3	2.1	25.9	20.4	55.1	16.6	35.4	25.2	52.6	33.2	43.4
4	-11.4	32.3	3.8	58.9	5.8	49.8	21.6	54.3	26.2	73.2
5	1.2	27.4	37.3	24.3	27.7	52.2	64.5	41.3	63.9	31.8
6	5.7	26.0	34.3	33.1	30.2	45.5	26.9	60.5	34.2	48.7
7	8.9	24.6	15.5	33.7	39.3	37.4	32.4	42.4	43.6	34.7
8	-7	28.3	9.3	44.6	22.8	39.2	17.0	46.2	37.0	40.0
9	-14.6	20.5	9.6	40.3	18.2	27.1	30.6	30.9	20.5	38.7
10	-9.1	29.2	9.6	40.3	18.2	27.1	30.2	30.9	20.5	38.7
11	13.7	26.9	16.3	55.4	13.4	37.9	30.2	46.6	20.7	47.5
12	-2.3	23.0	30.5	53.4	15.6	49.3	12.9	54.4	27.3	48.7
13	-1.6	25.7	6.8	49.6	5.0	59.6	23.5	62.1	11.4	62.9
14	-5.6	25.9	29.5	37.5	14.7	35.4	21.4	55.6	34.0	40.6
15	2.9	25.6	1.9	39.1	9.2	59.9	23.7	61.5	10.5	43.9
16	-5	25.2	14.5	49.8	9.7	50.4	8.9	55.5	-5.9	64.5

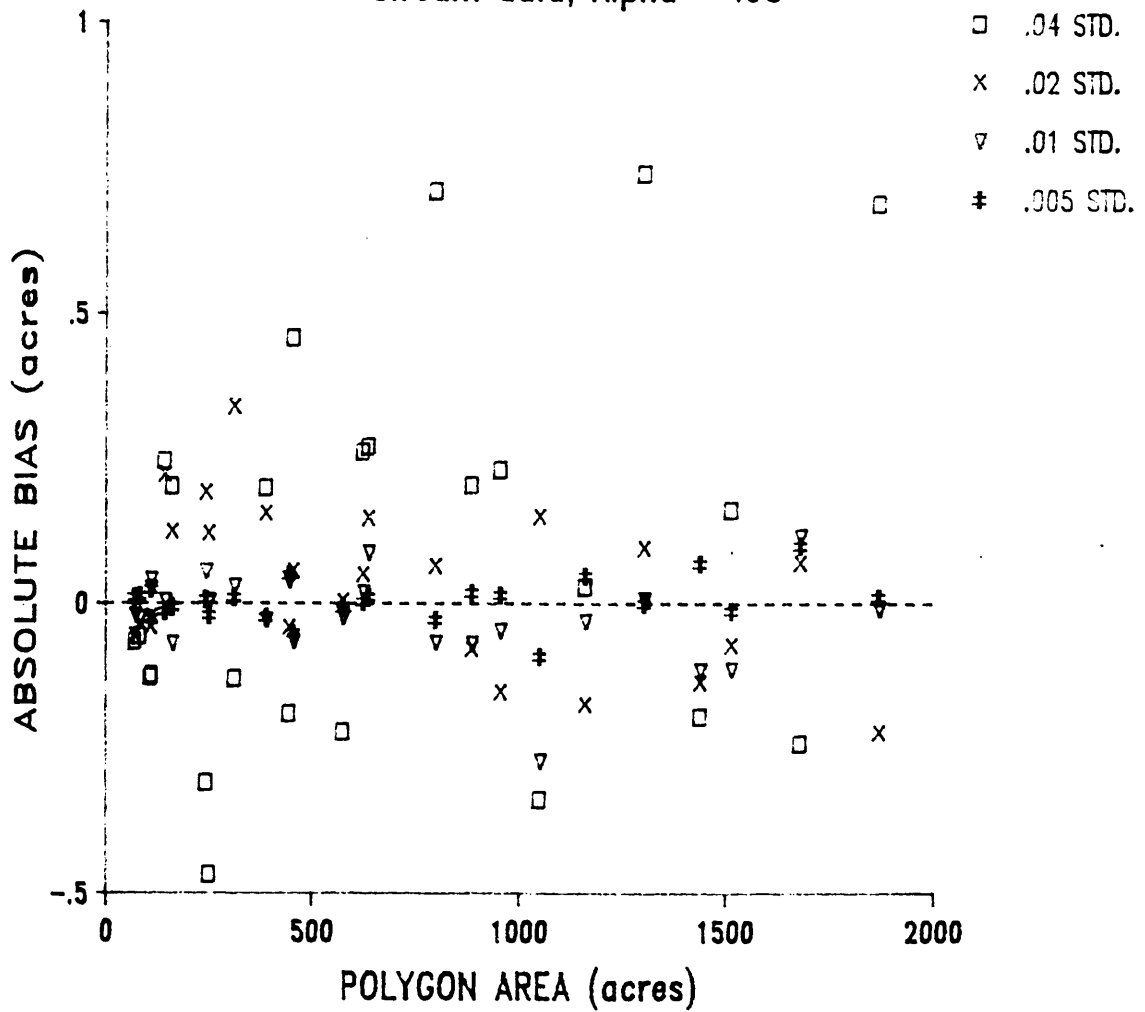
Point Mode Digitizing

Test Feature	operator 1		operator 2		operator 3		operator 4		operator 5	
	Mean .001 in	Std.Dev. .001 in	Mean .001 in	Std.Dev. .001 in	Mean .001 in	Std.Dev. .001 in	Mean .001 in	Std.Dev. .001 in	Mean .001 in	Std.Dev. .001 in
1	-12.1	37.2	-13.4	26.7	.5	33.7	-5.5	33.7	7.0	42.4
2	10.3	25.5	.7	24.6	17.7	27.5	23.5	28.8	17.0	29.9
3	6.1	18.8	35.0	46.6	20.3	31.5	18.4	33.0	12.5	35.6
4	-6	18.8	19.1	61.3	9.0	44.6	19.7	54.5	15.6	33.6
5	10.2	19.7	49.0	16.9	35.2	34.9	37.9	33.6	40.5	36.1
6	-3.6	19.8	48.3	39.1	32.2	38.8	13.9	43.1	26.9	27.2
7	7.3	19.6	44.2	23.6	33.5	21.0	21.8	26.3	28.6	25.6
8	2.5	18.7	21.9	43.9	15.5	41.6	20.5	33.4	26.3	33.0
9	-3.6	19.8	14.0	38.2	15.9	40.1	10.8	31.6	7.1	34.9
10	-5.3	20.1	12.4	37.1	-1.1	33.4	6.9	35.0	8.1	31.8
11	12.1	23.2	44.4	45.9	31.5	35.6	39.6	34.3	29.1	40.6
12	4.8	21.4	32.4	35.1	22.2	36.9	23.2	43.6	10.0	31.1
13	-3.8	19.3	-6.3	50.7	15.2	46.4	3.5	45.1	7.0	43.2
14	5.6	20.7	24.4	36.6	30.2	27.6	21.2	43.2	25.6	29.9
15	11.2	36.8	41.3	43.8	18.3	47.6	50.9	70.0	28.6	48.6
16	-4.0	21.7	11.4	38.4	-8	56.4	16.2	38.7	6.1	34.4

Appendix D. Graphs of Simulation Results

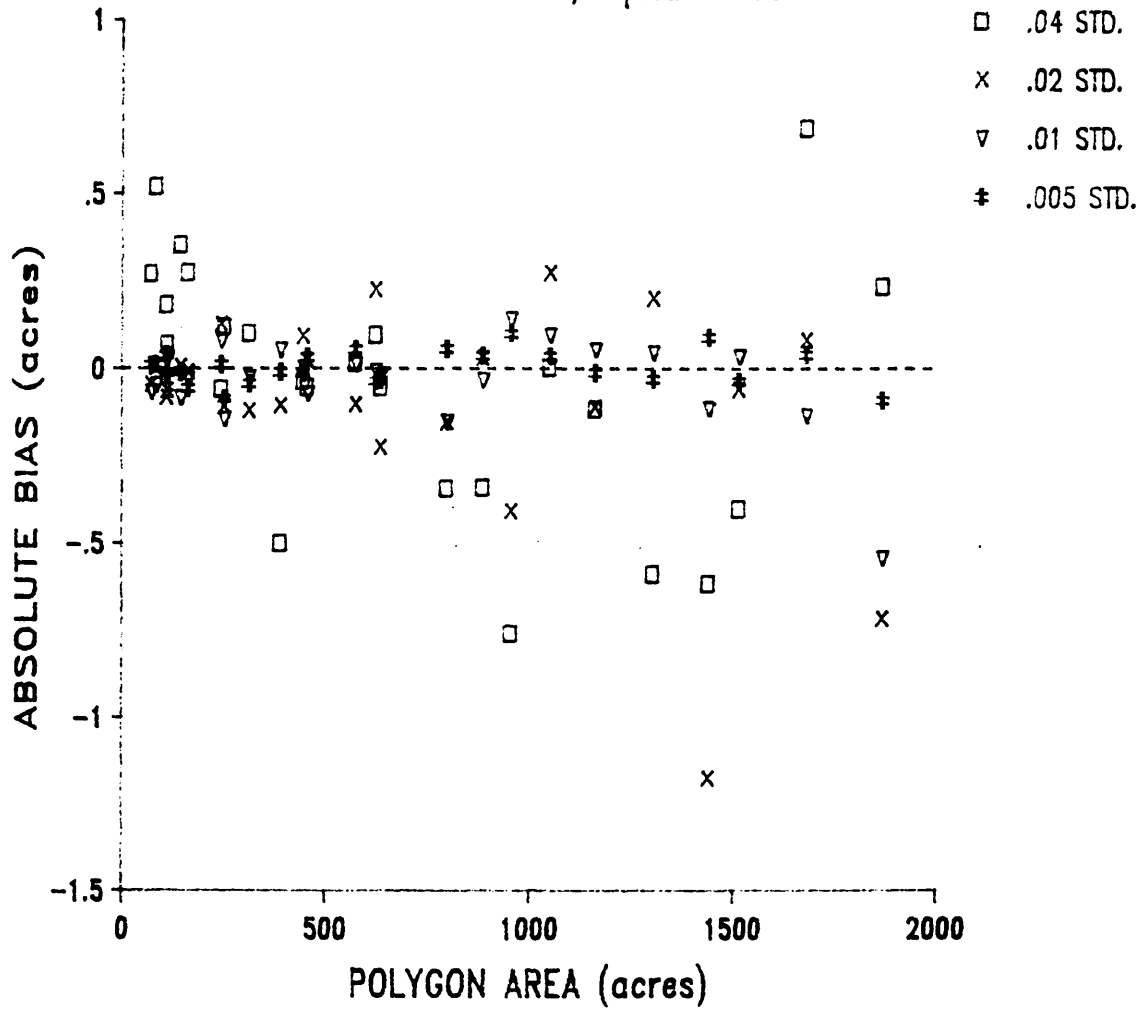
ABSOLUTE BIAS VS. POLYGON AREA

Stream data, Alpha = .05



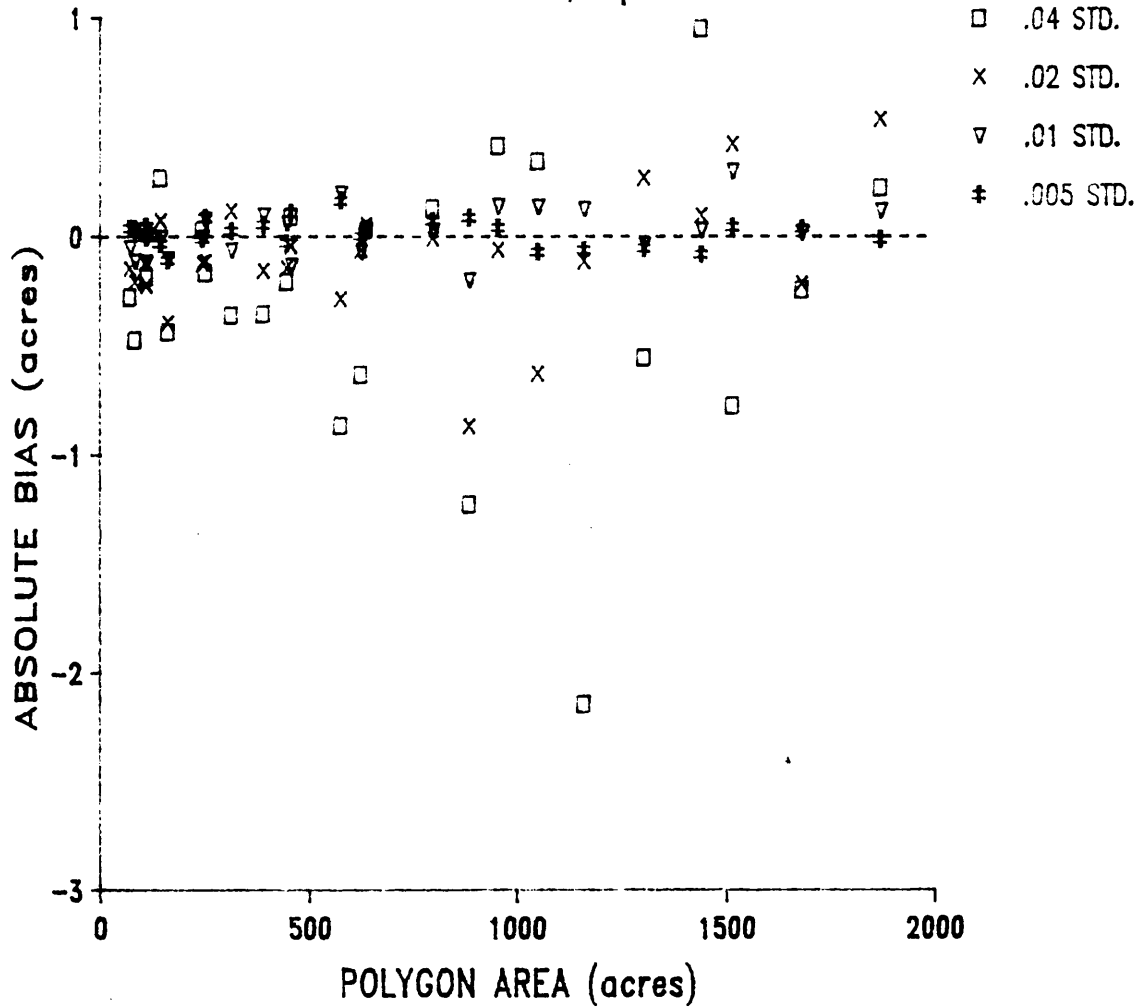
ABSOLUTE BIAS VS. POLYGON AREA

Stream data, Alpha = .10



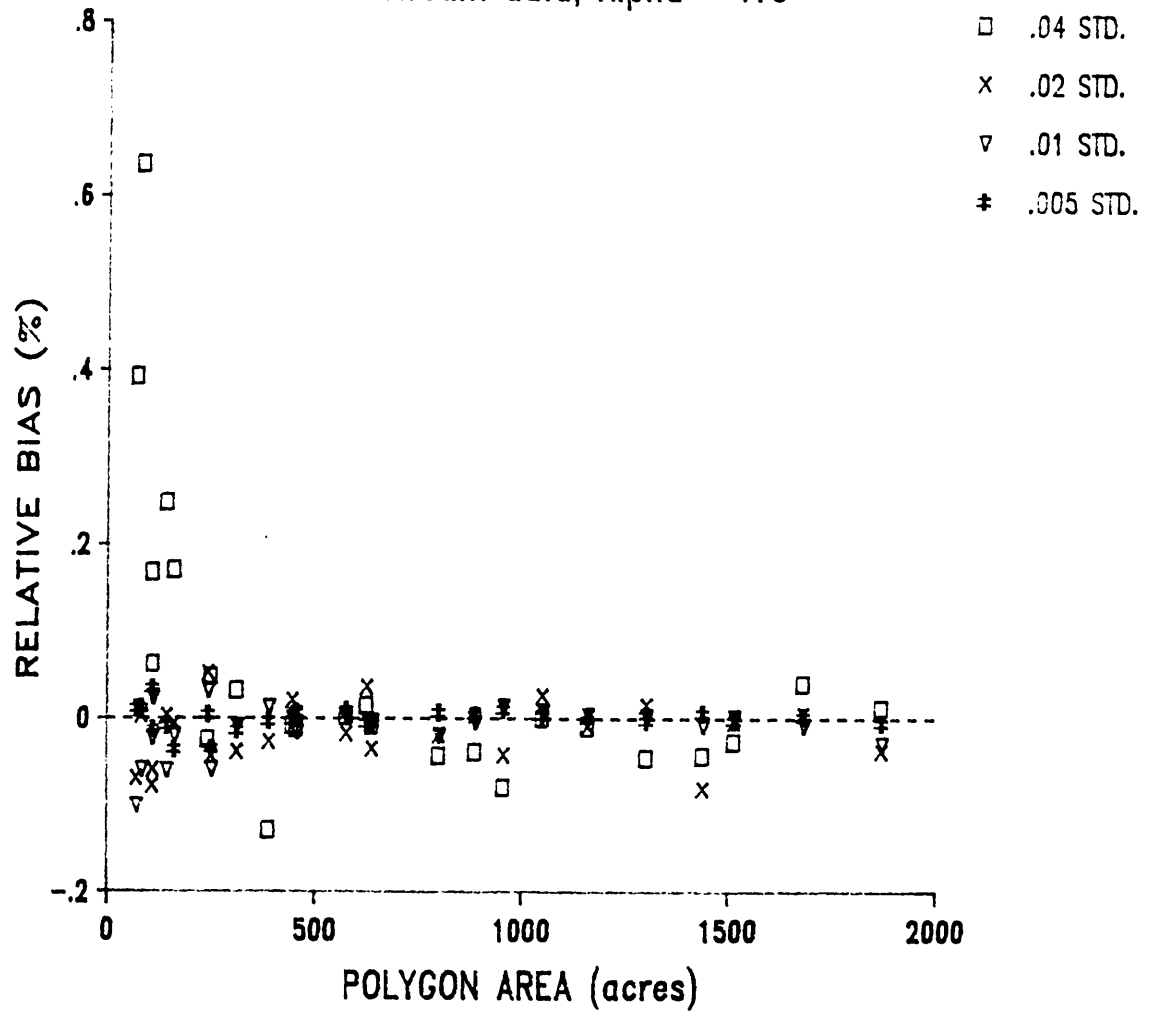
ABSOLUTE BIAS VS. POLYGON AREA

Stream data, Alpha = .20



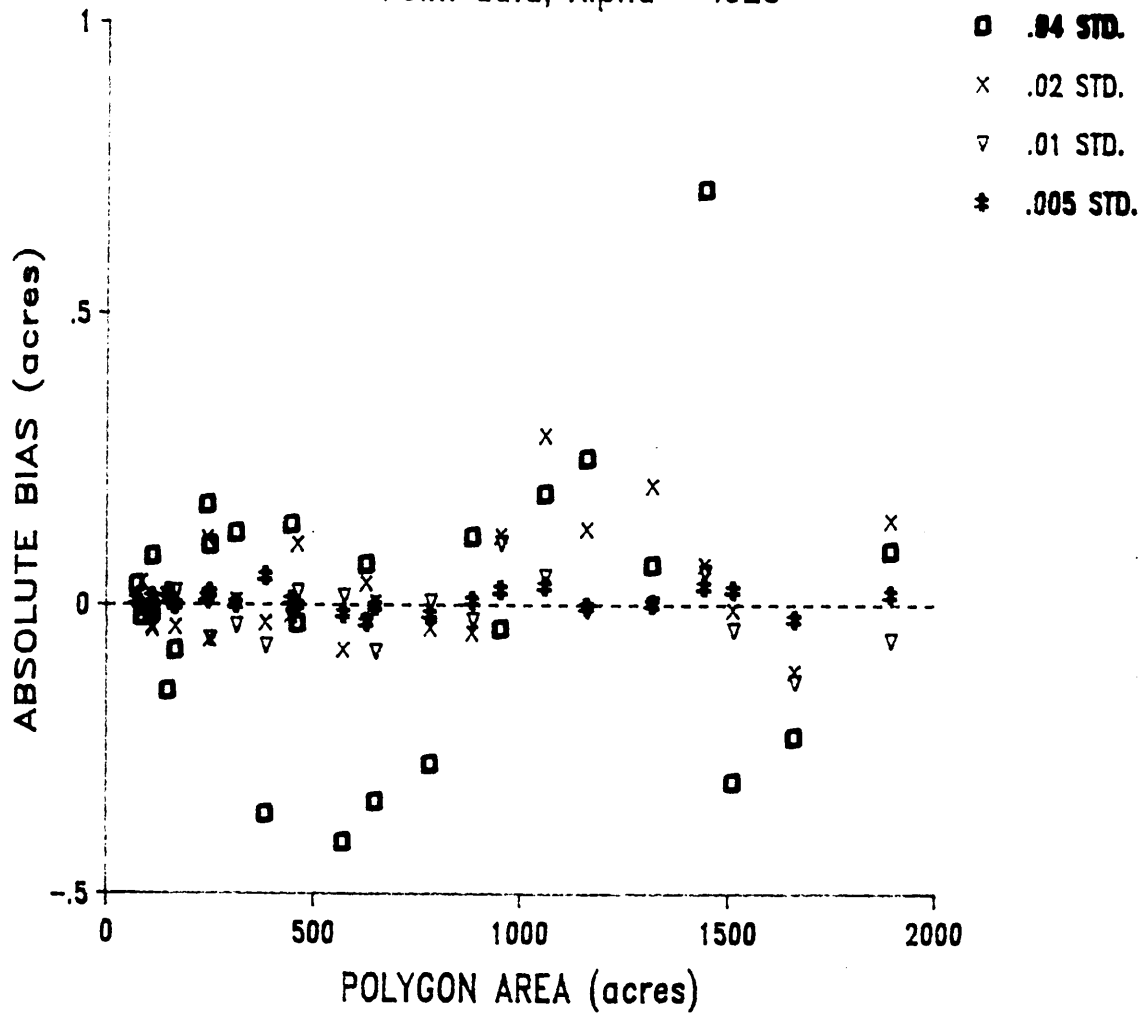
RELATIVE BIAS VS. POLYGON AREA

Stream data, Alpha = .10



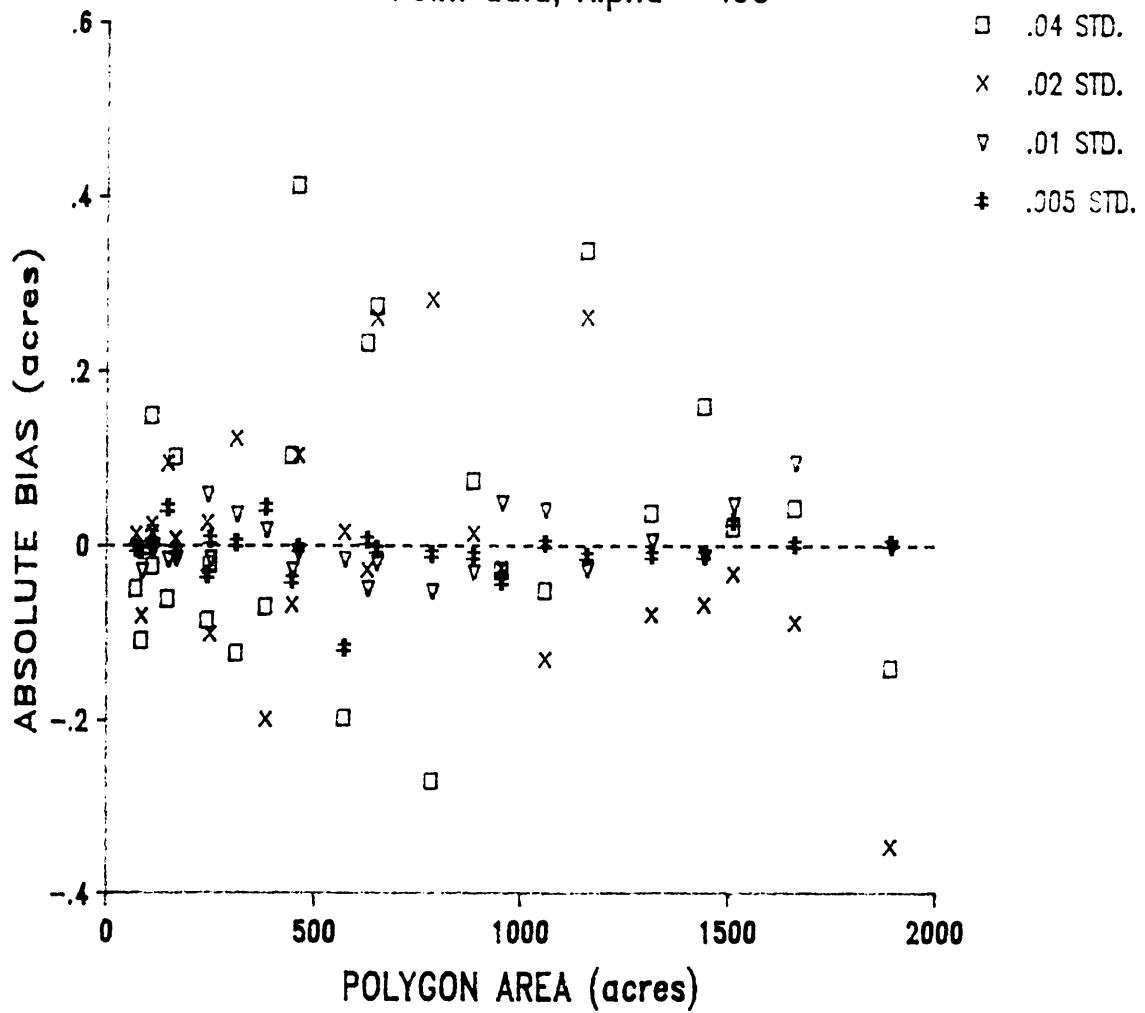
ABSOLUTE BIAS VS. POLYGON AREA

Point data, Alpha = .025



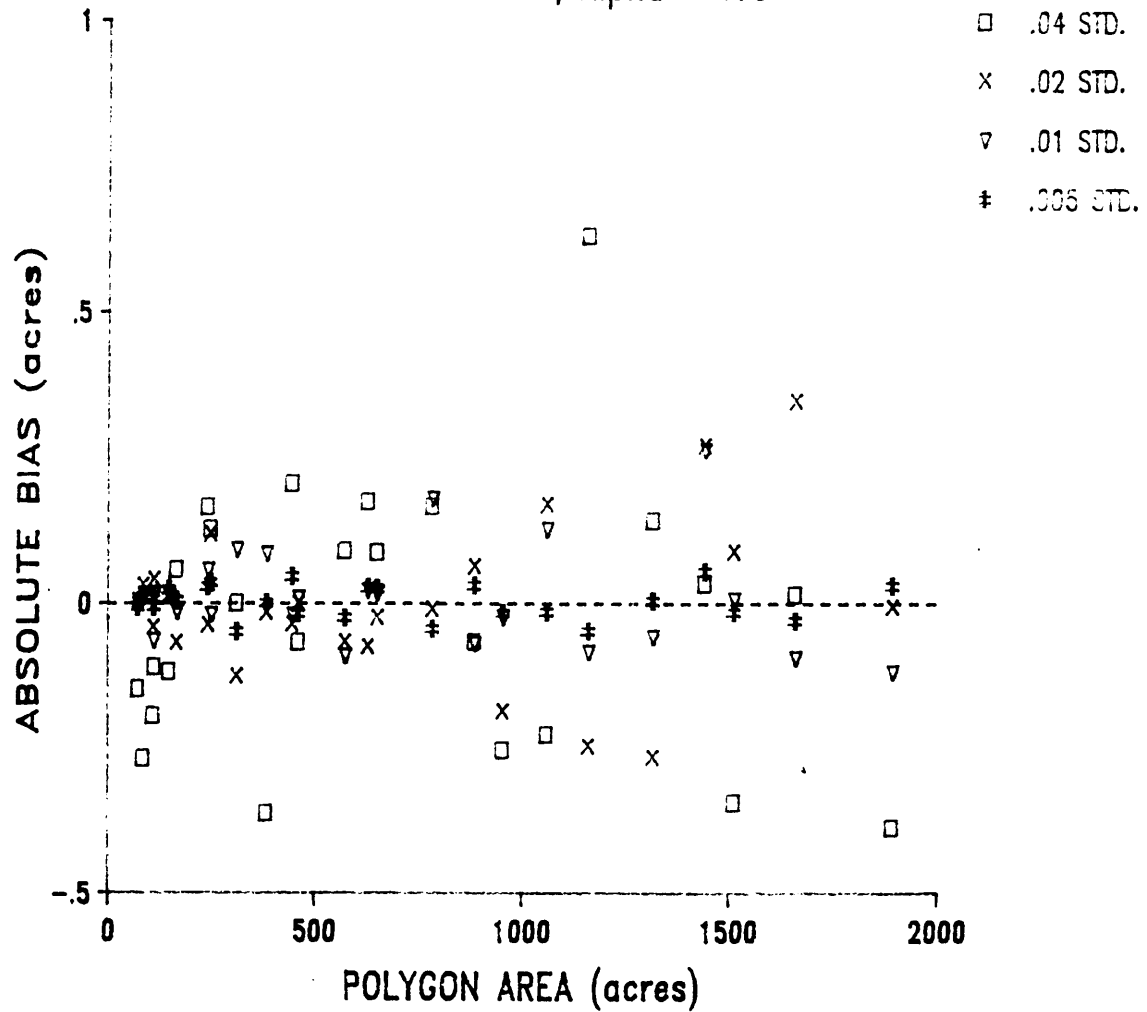
ABSOLUTE BIAS VS. POLYGON AREA

Point data, Alpha = .05



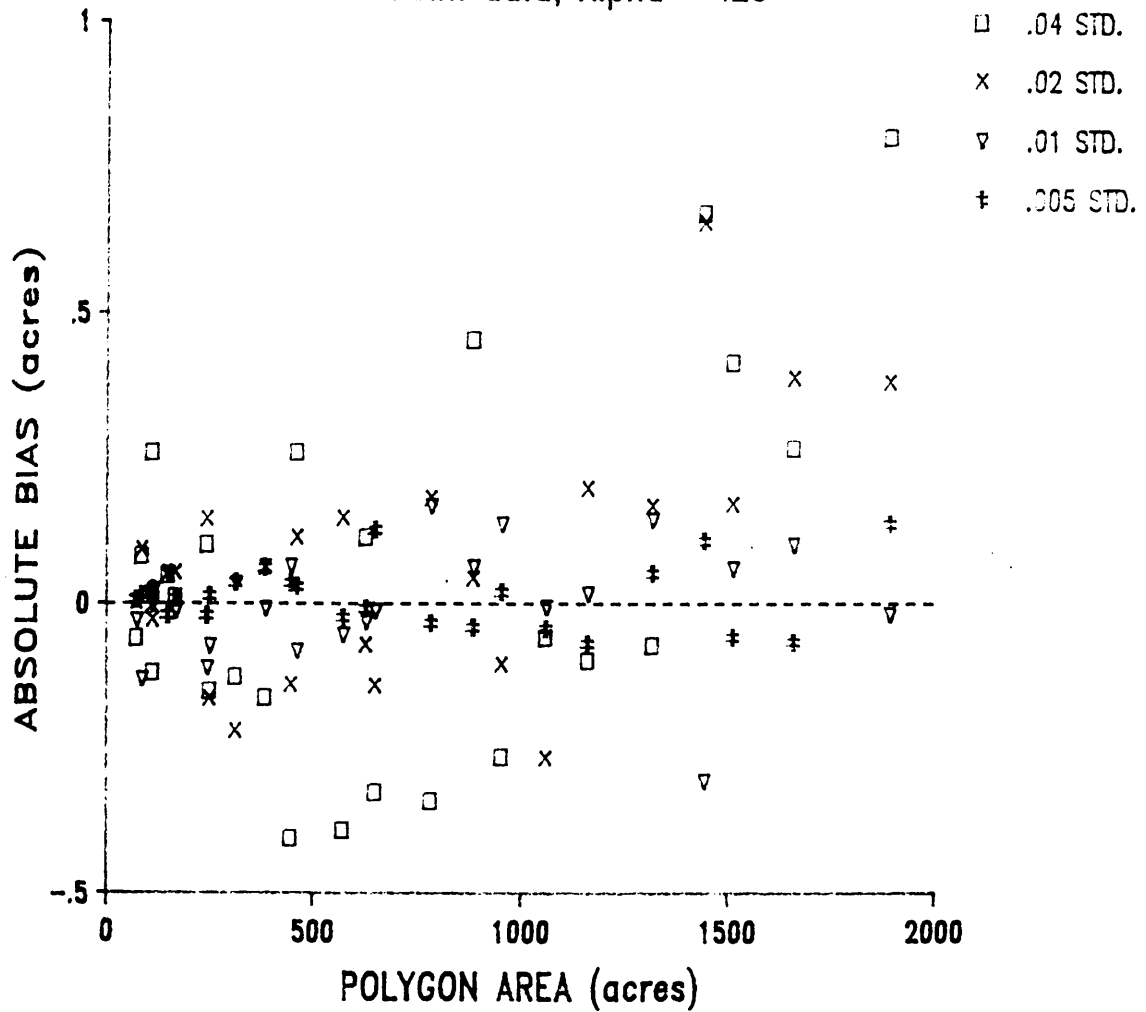
ABSOLUTE BIAS VS. POLYGON AREA

Point data, Alpha = .10



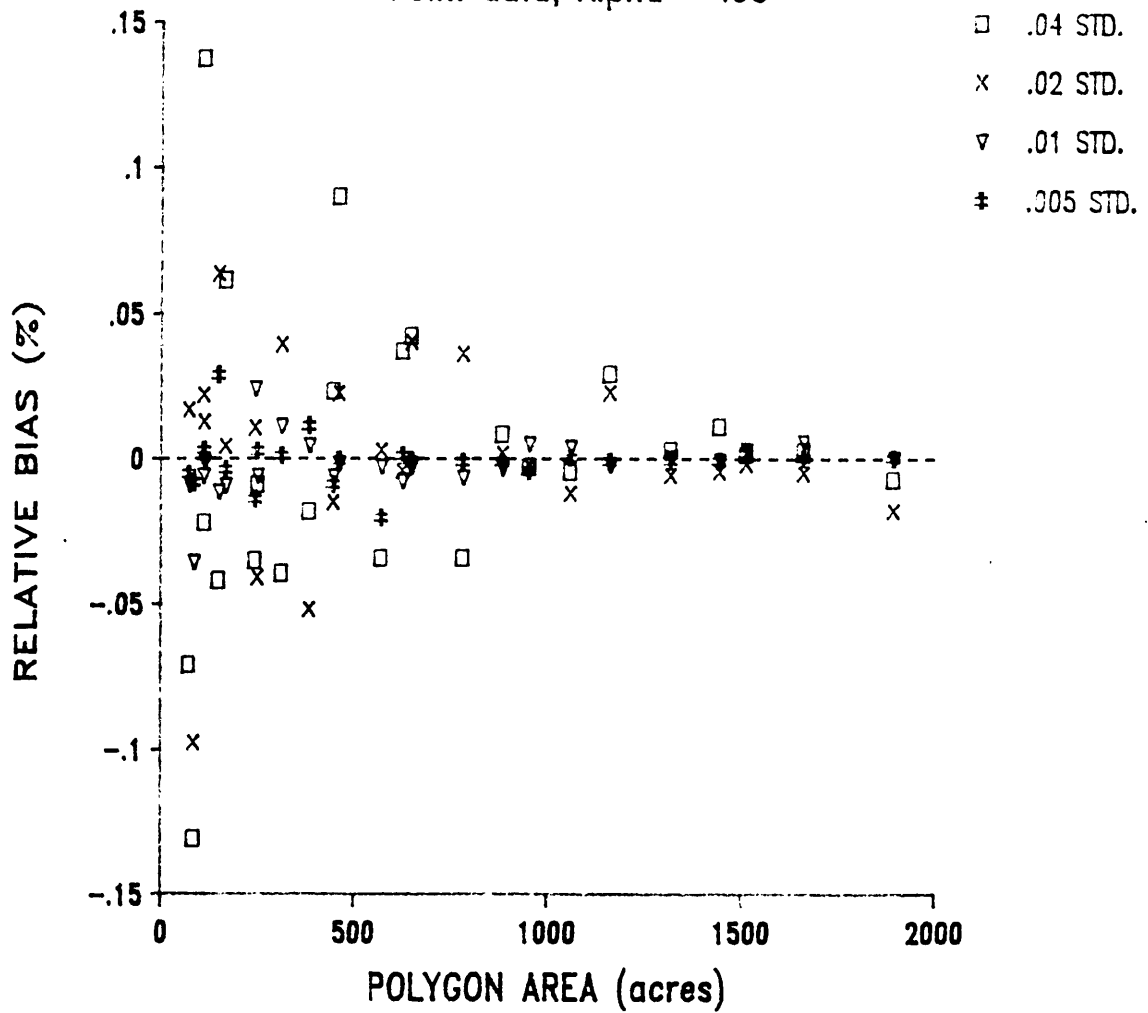
ABSOLUTE BIAS VS. POLYGON AREA

Point data, Alpha = .20



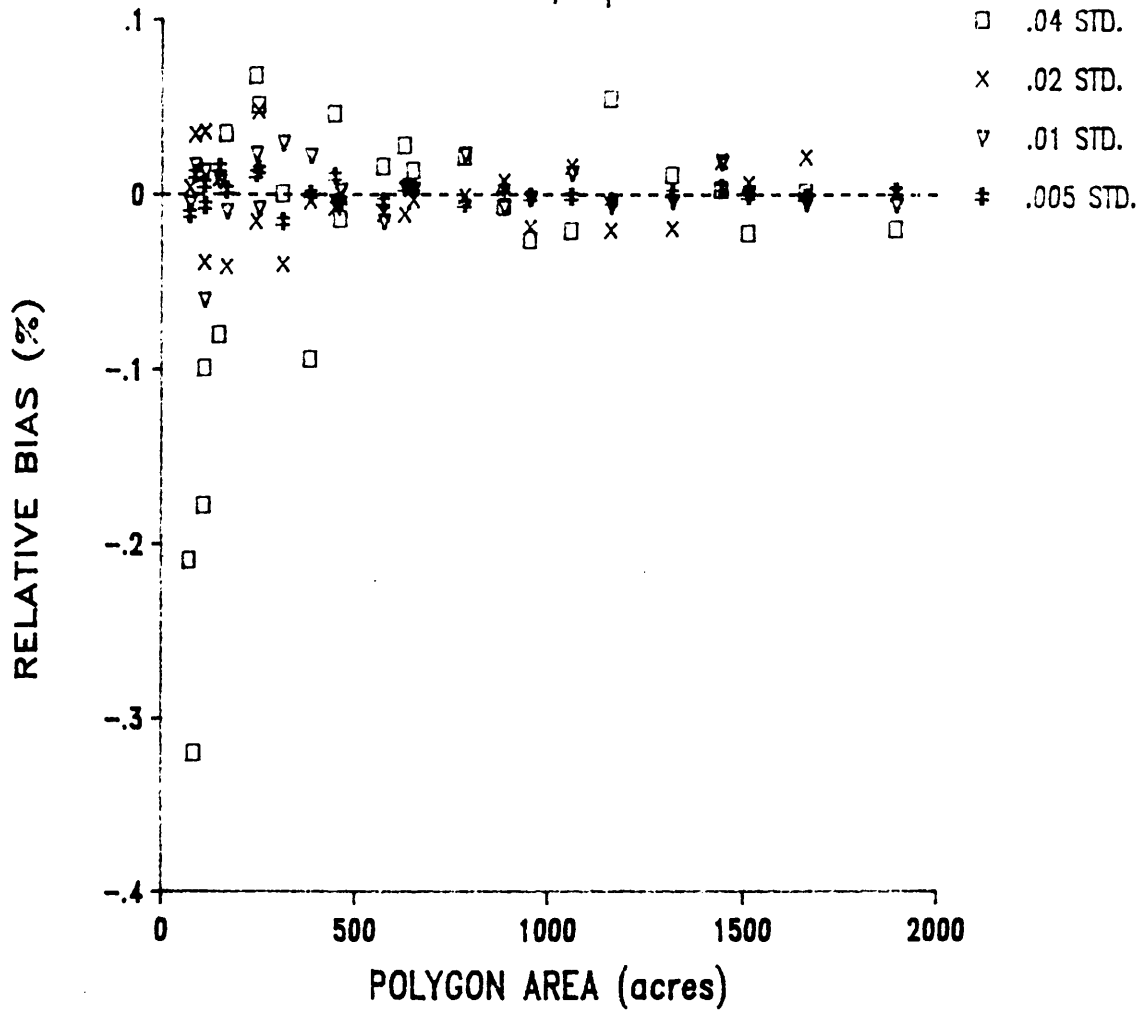
RELATIVE BIAS VS. POLYGON AREA

Point data, Alpha = .05



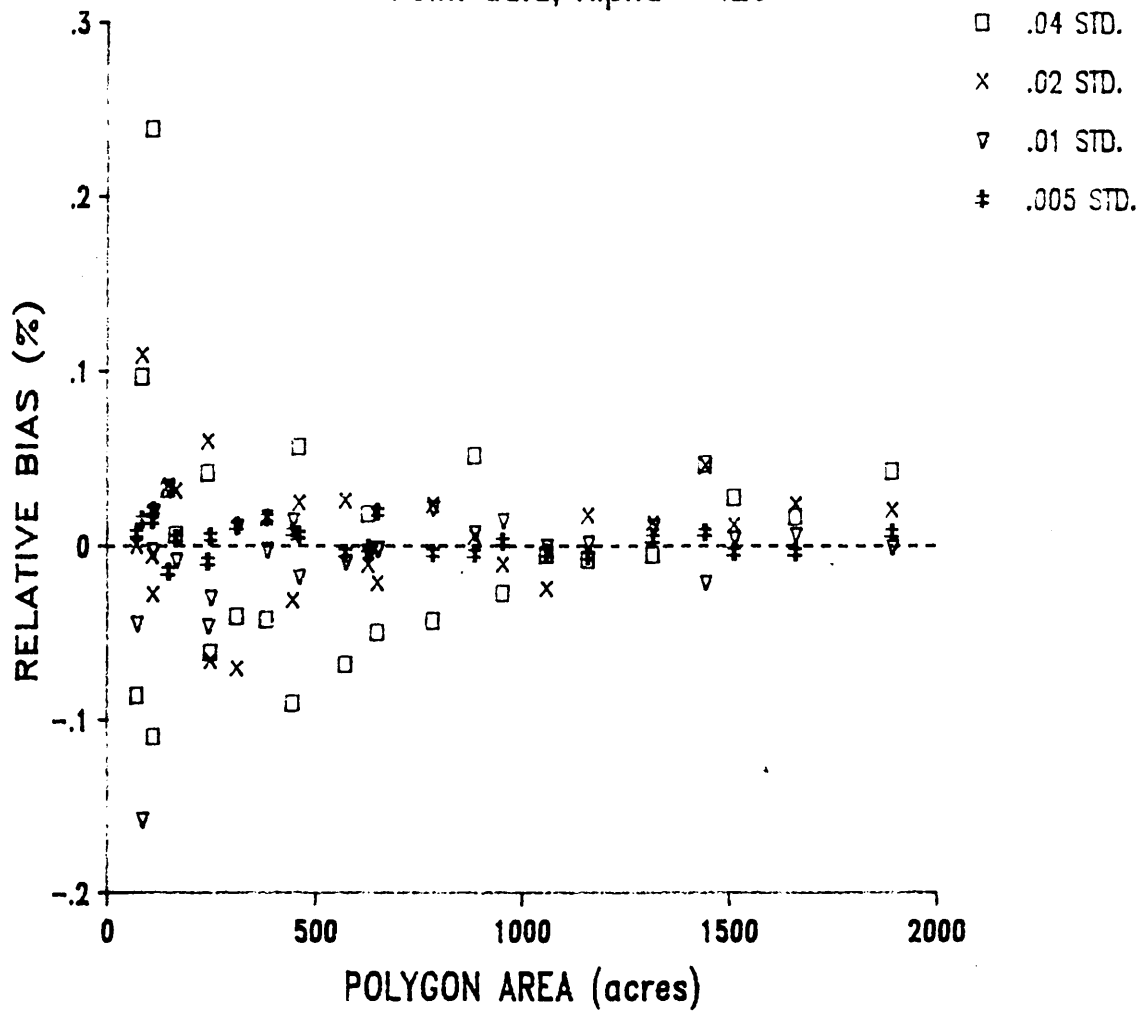
RELATIVE BIAS VS. POLYGON AREA

Point data, Alpha = .10



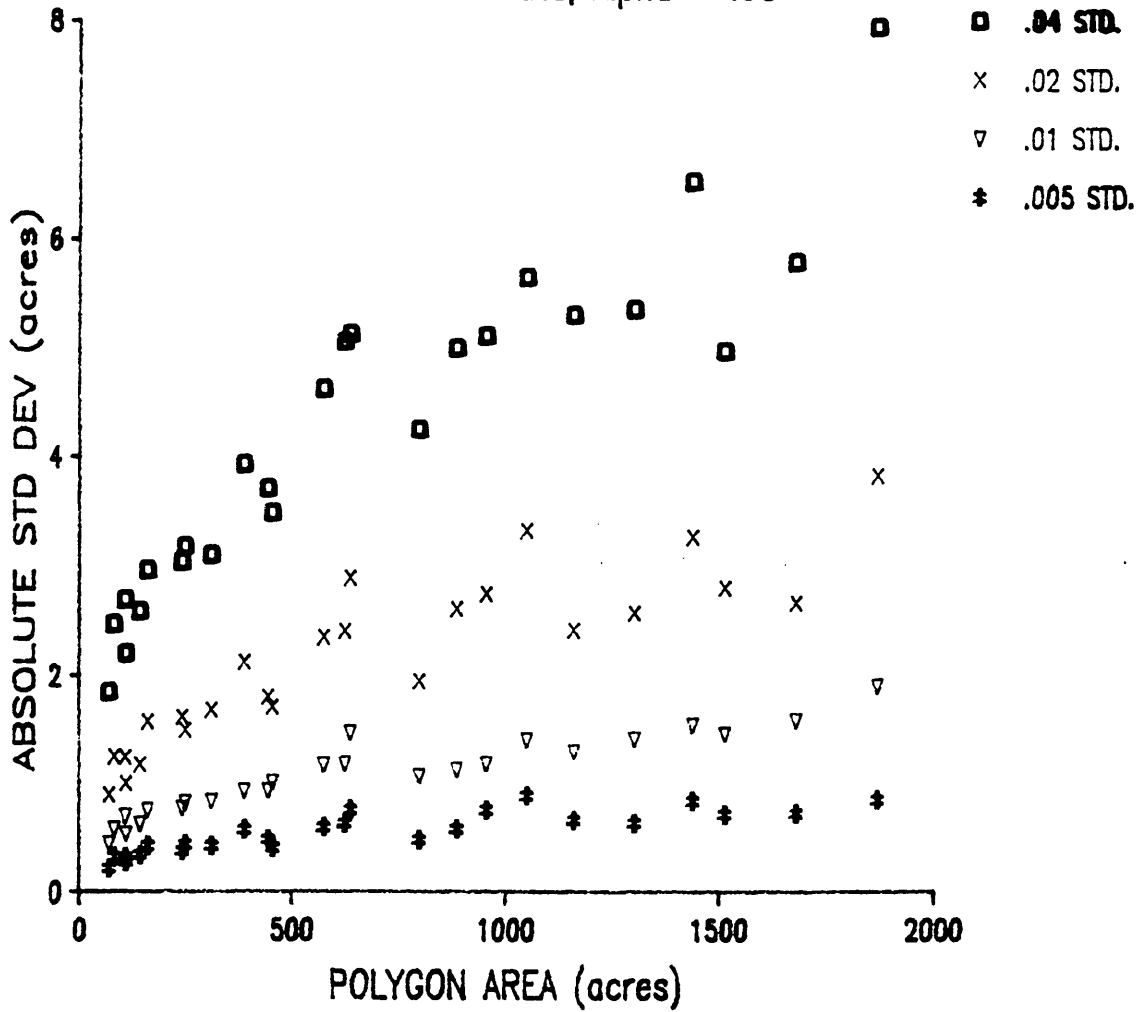
RELATIVE BIAS VS. POLYGON AREA

Point data, Alpha = .20



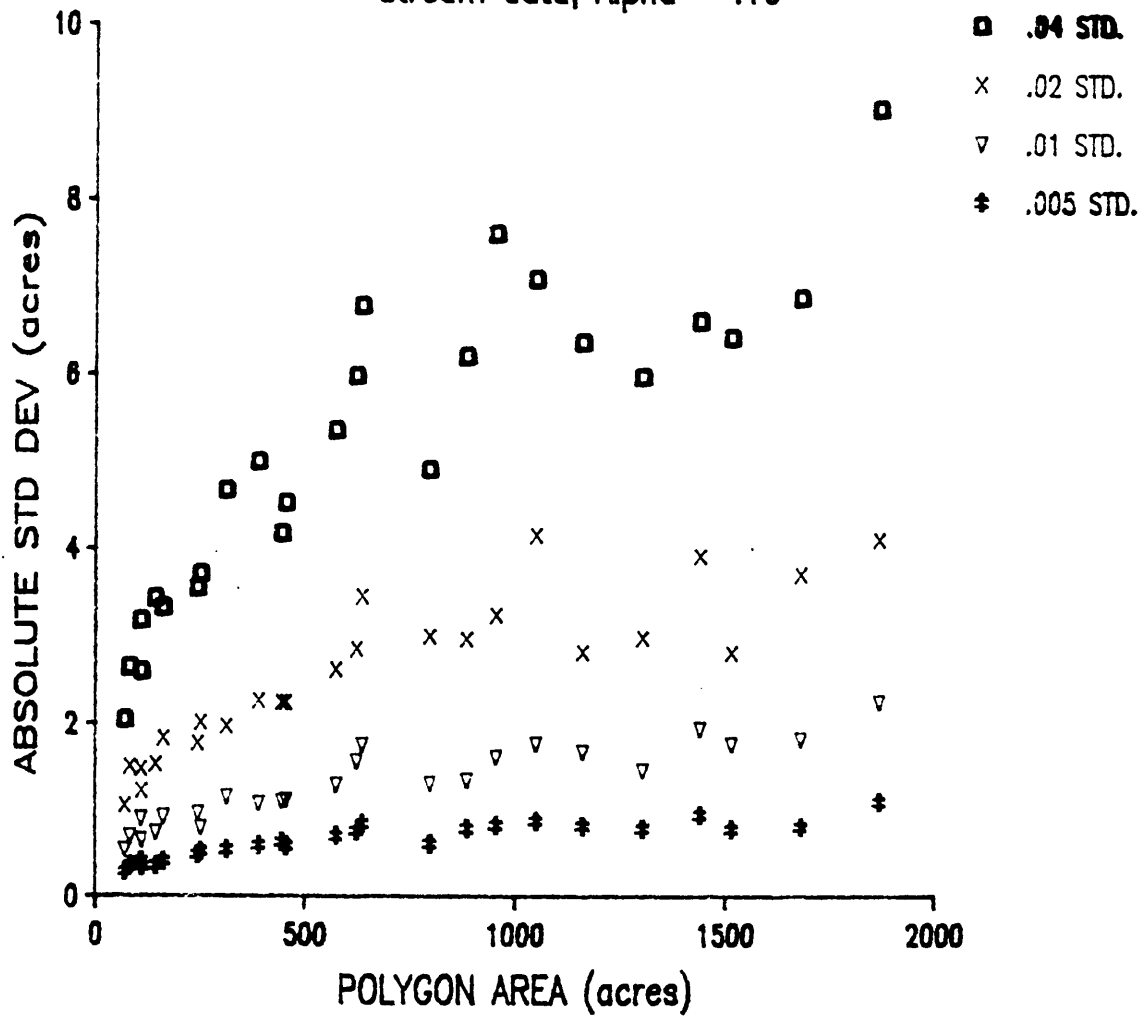
ABSOLUTE STD DEV VS. POLYGON AREA

Stream data, Alpha = .05



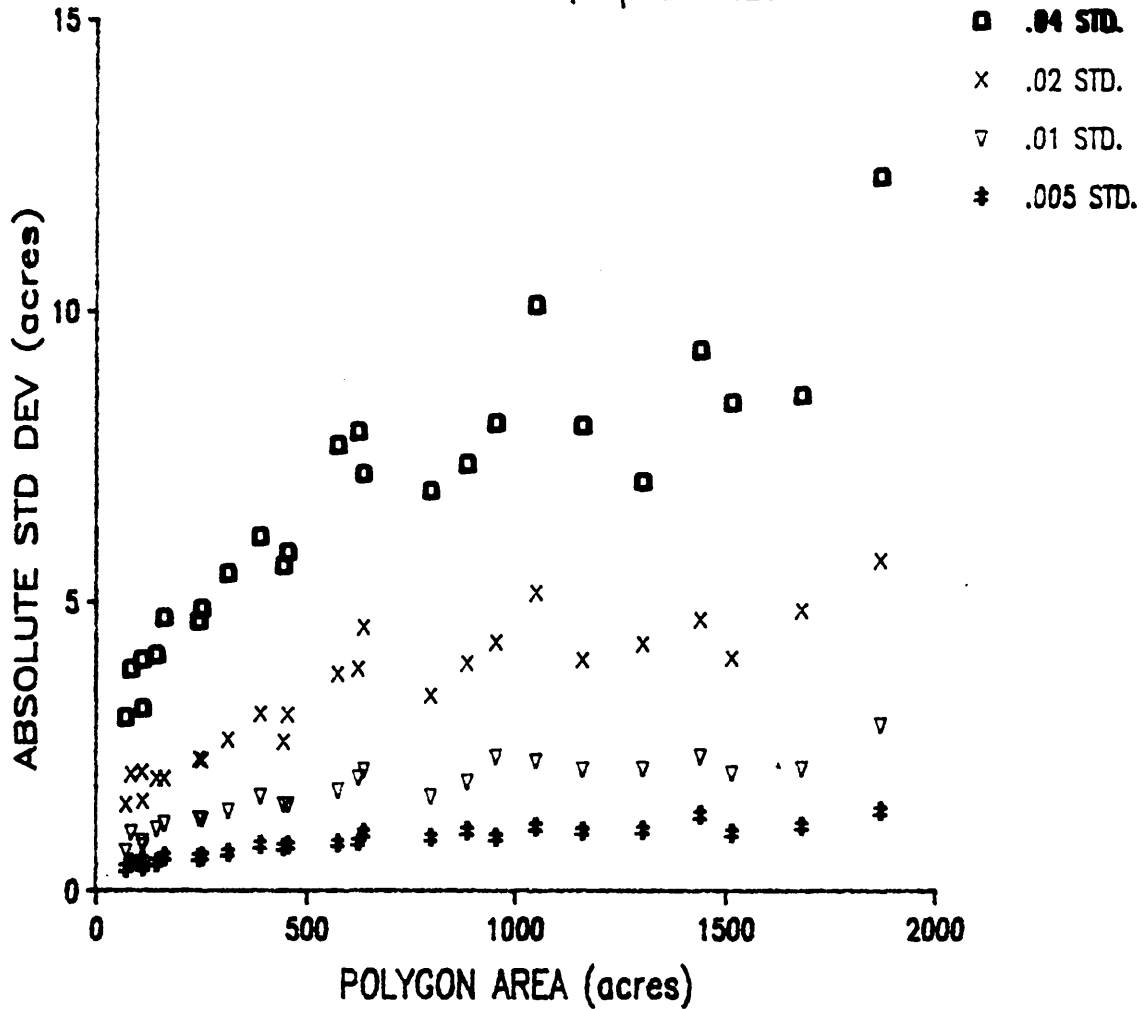
ABSOLUTE STD DEV VS. POLYGON AREA

Stream data, Alpha = .10



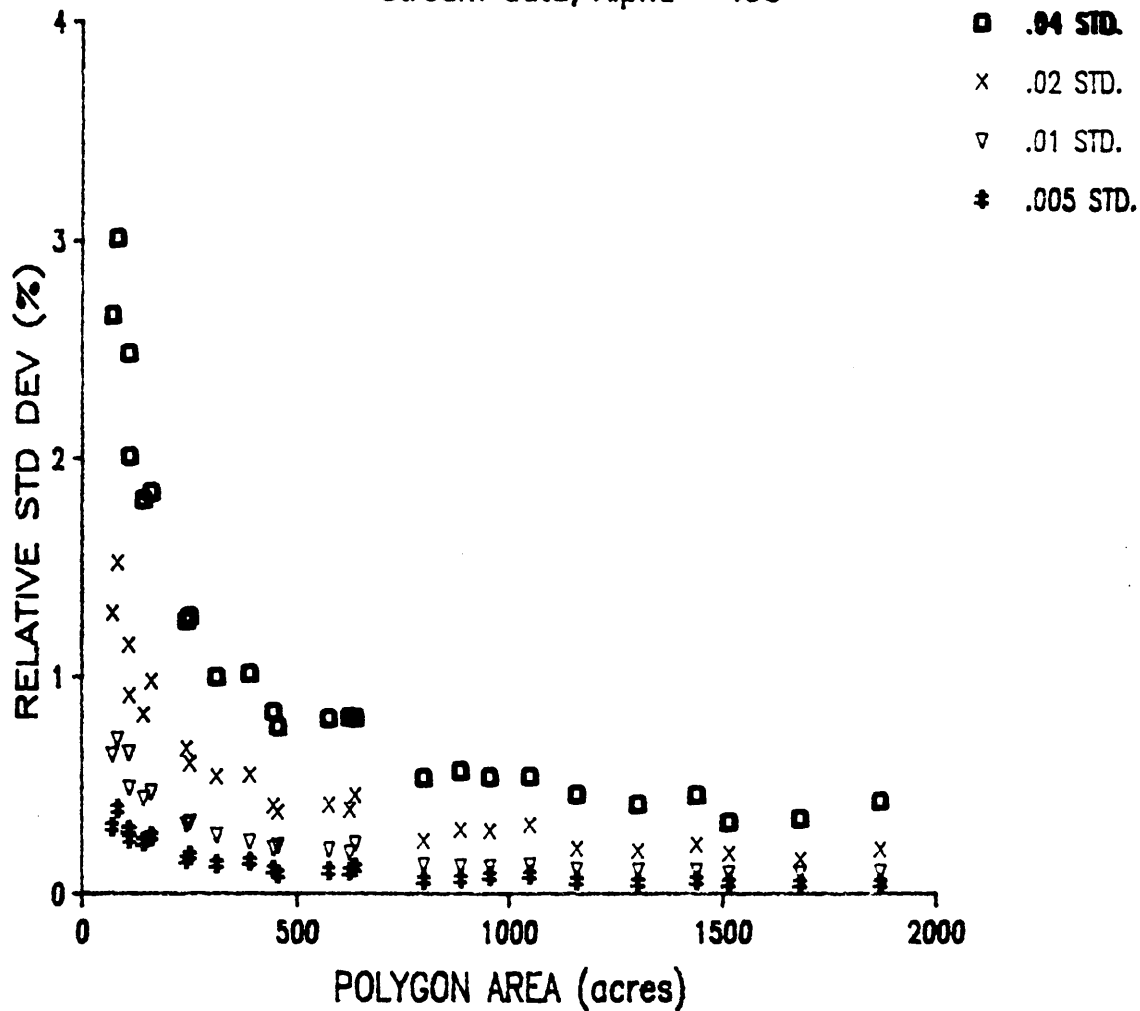
ABSOLUTE STD DEV VS. POLYGON AREA

Stream data, Alpha = .20



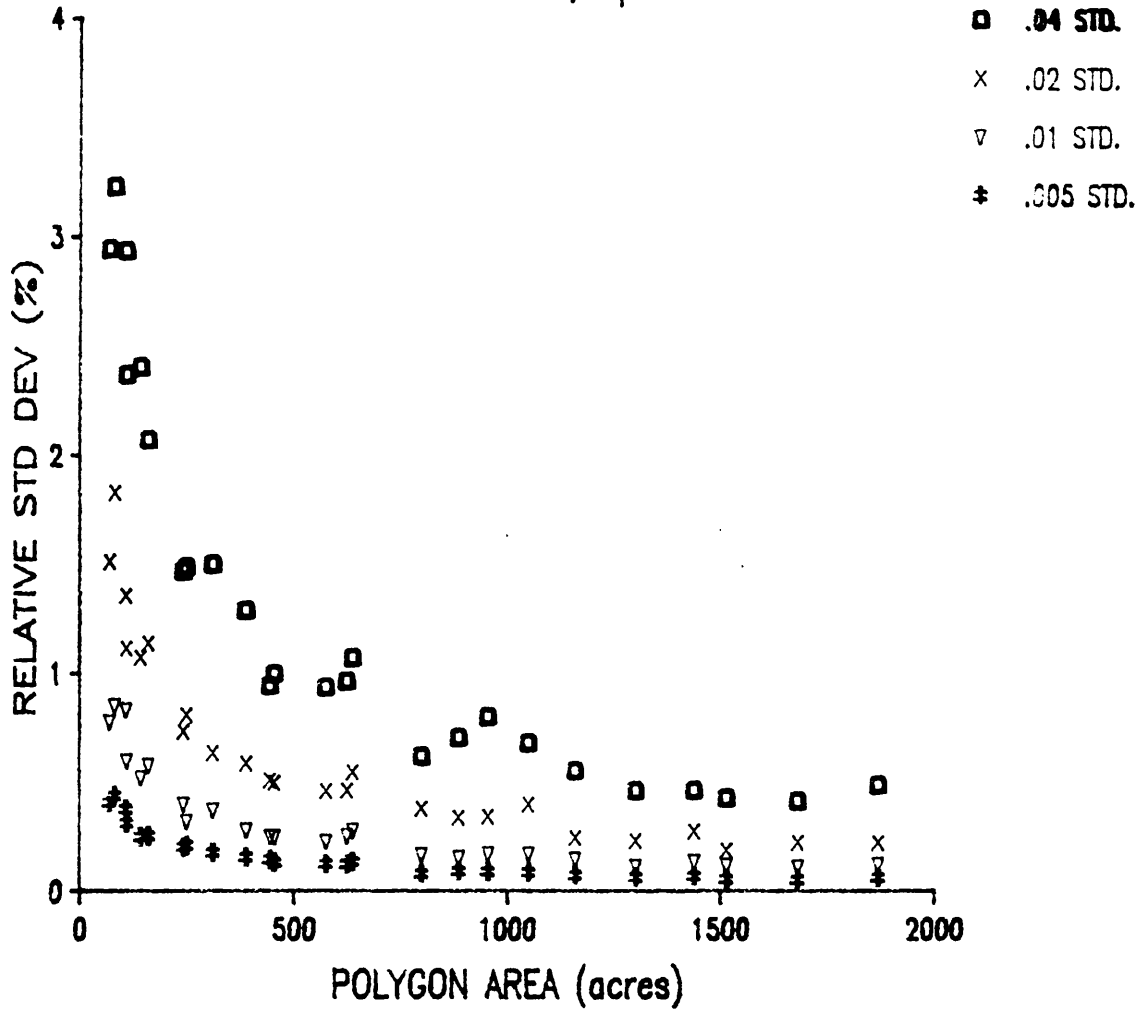
RELATIVE STD DEV VS. POLYGON AREA

Stream data, Alpha = .05



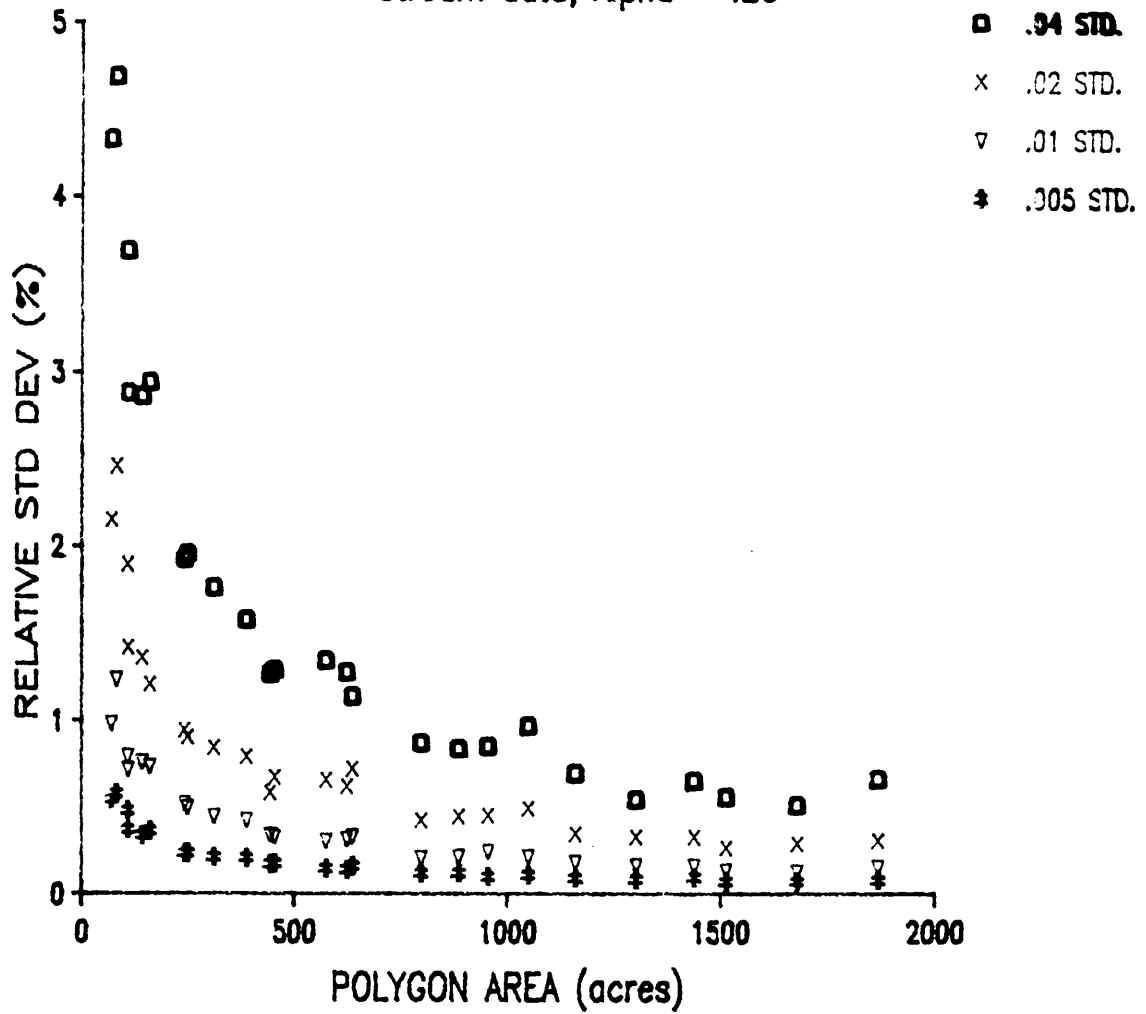
RELATIVE STD DEV VS. POLYGON AREA

Stream data, Alpha = .10



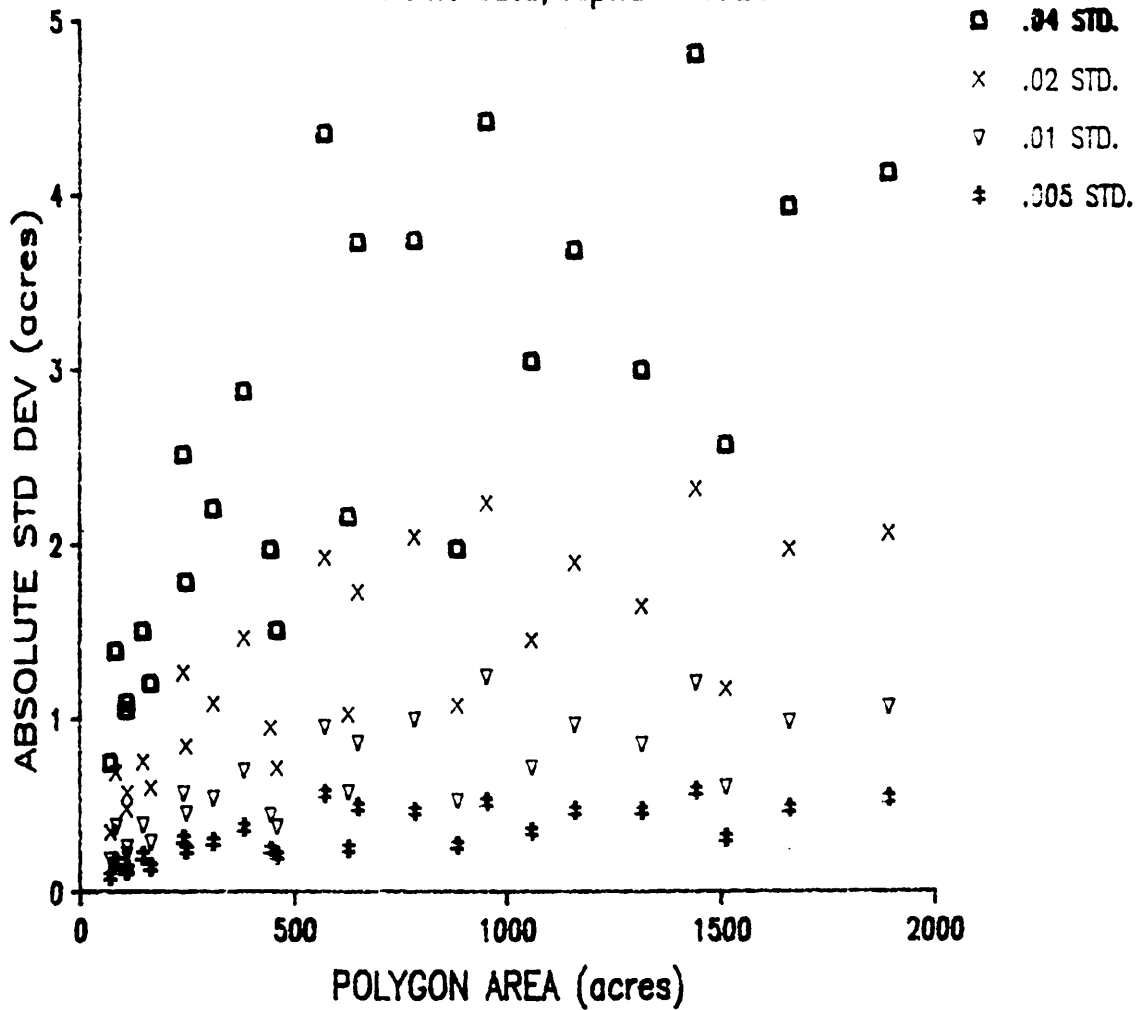
RELATIVE STD DEV VS. POLYGON AREA

Stream data, Alpha = .20



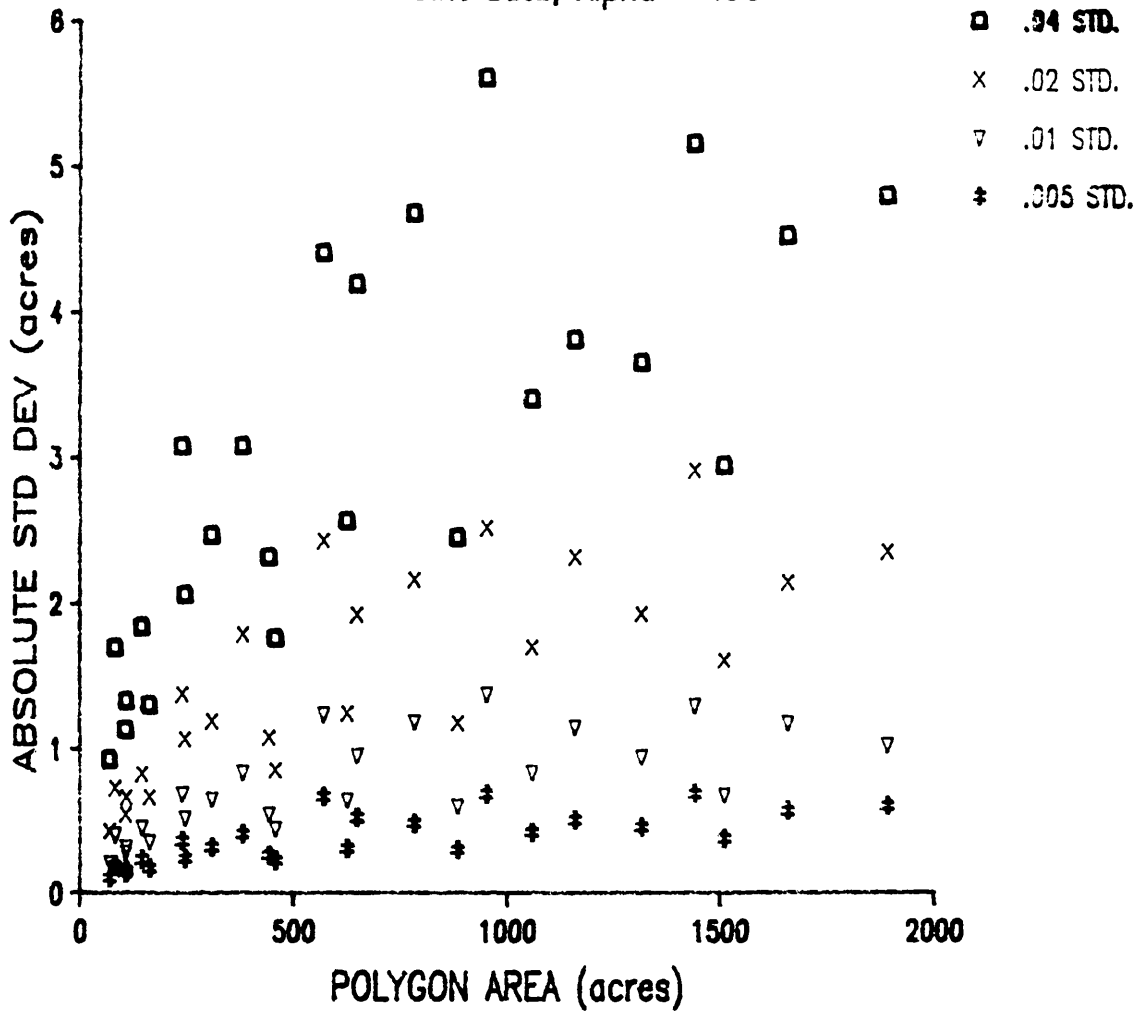
ABSOLUTE STD DEV VS. POLYGON AREA

Point data, Alpha = .025



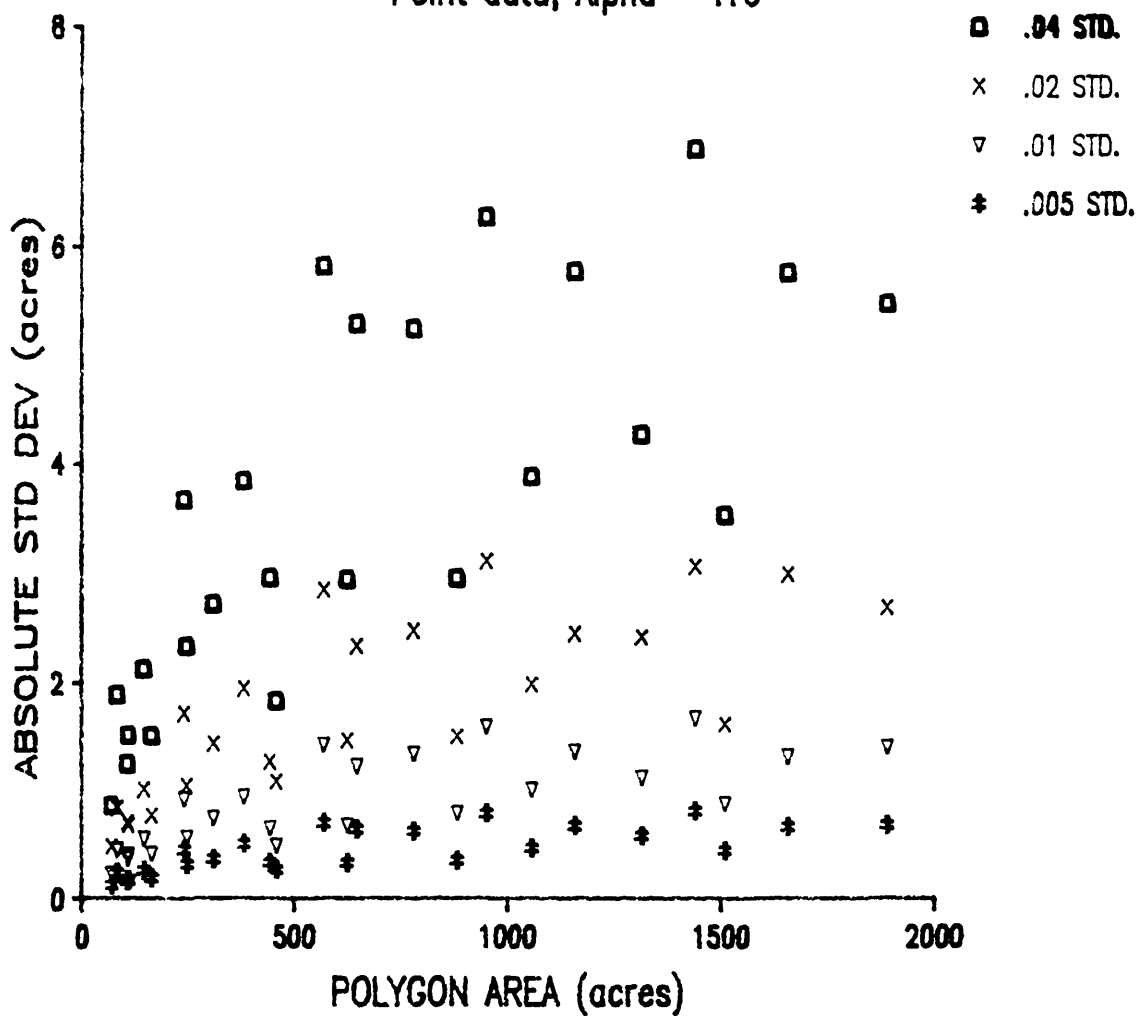
ABSOLUTE STD DEV VS. POLYGON AREA

Point data, Alpha = .05



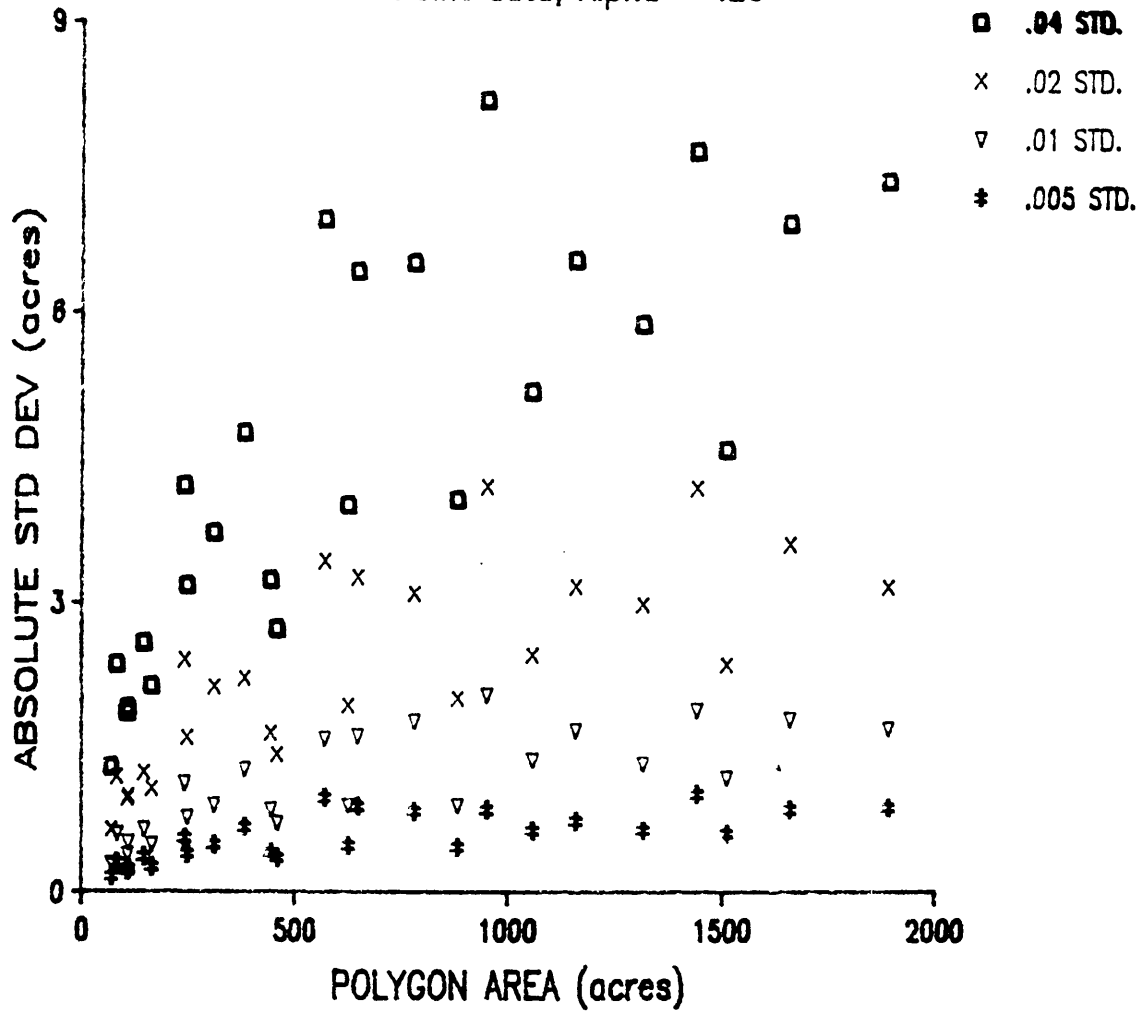
ABSOLUTE STD DEV VS. POLYGON AREA

Point data, Alpha = .10



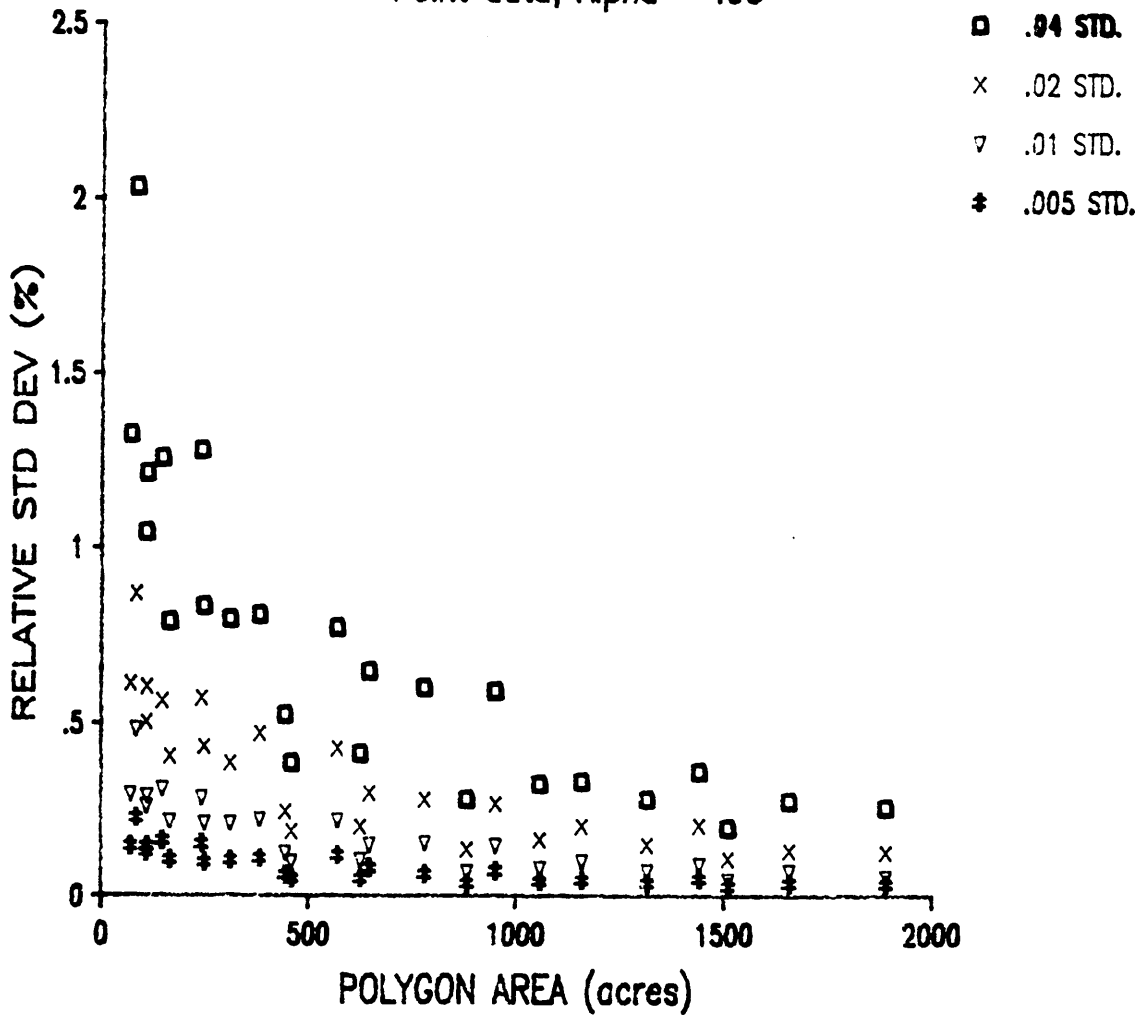
ABSOLUTE STD DEV VS. POLYGON AREA

Point data, Alpha = .20



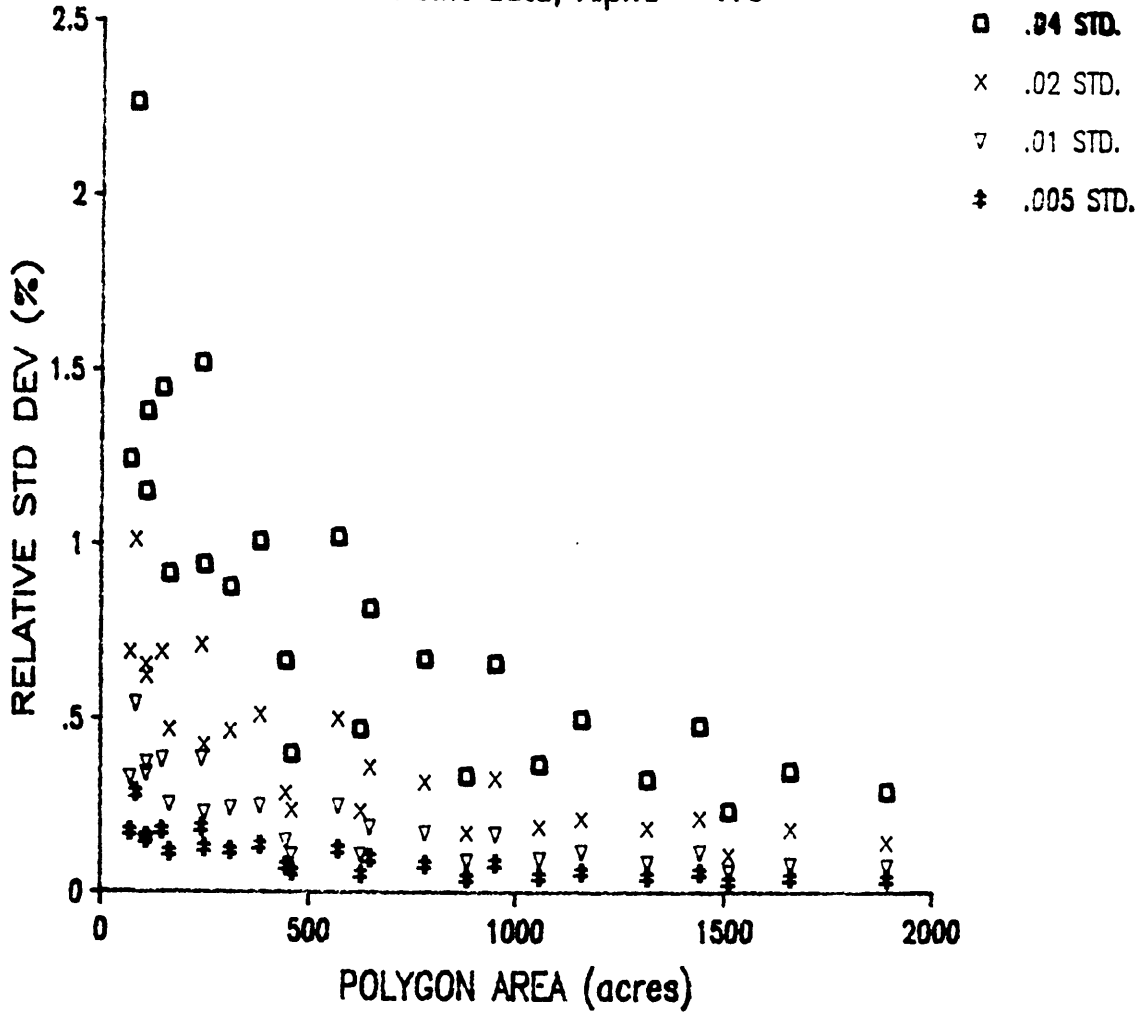
RELATIVE STD DEV VS. POLYGON AREA

Point data, Alpha = .05



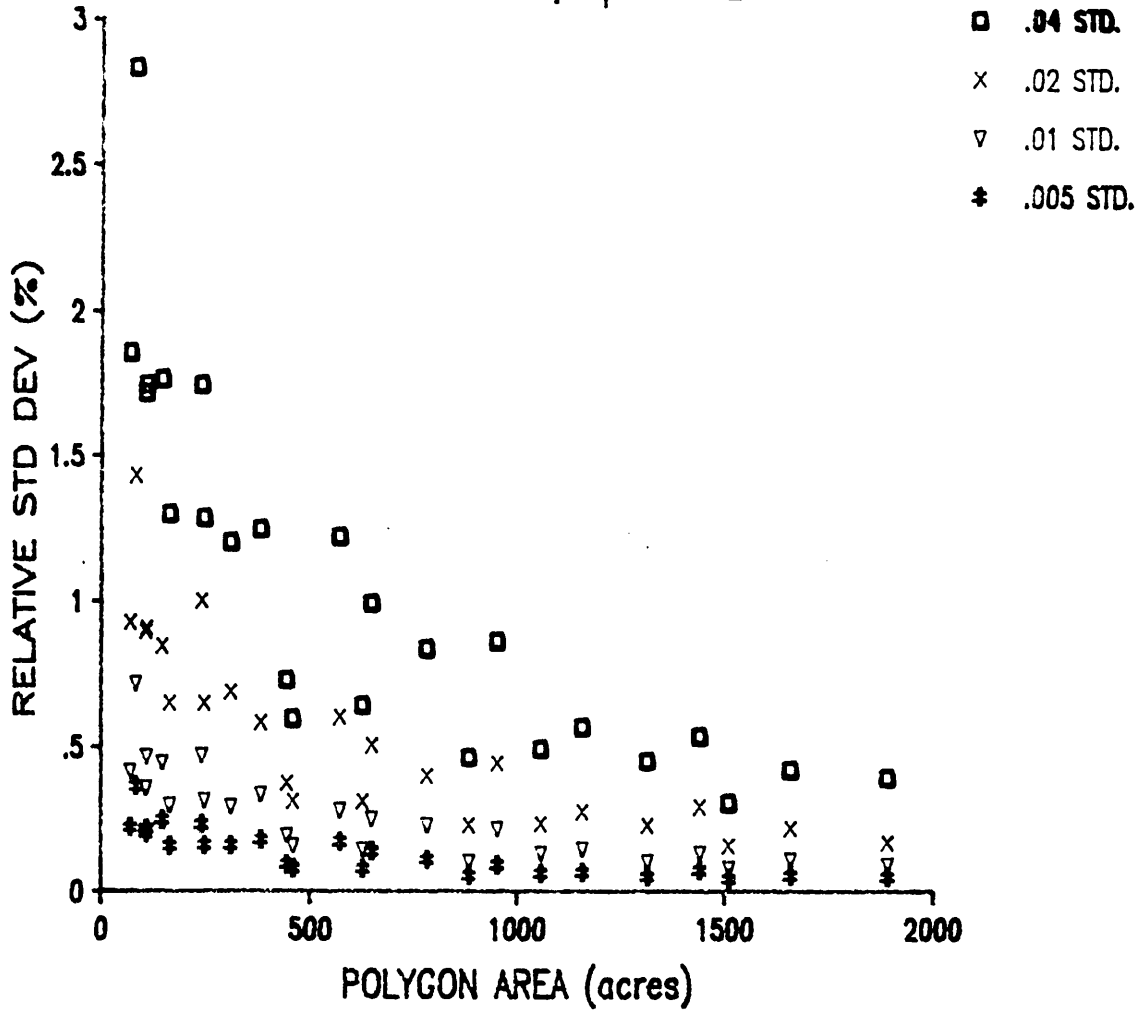
RELATIVE STD DEV VS. POLYGON AREA

Point data, Alpha = .10



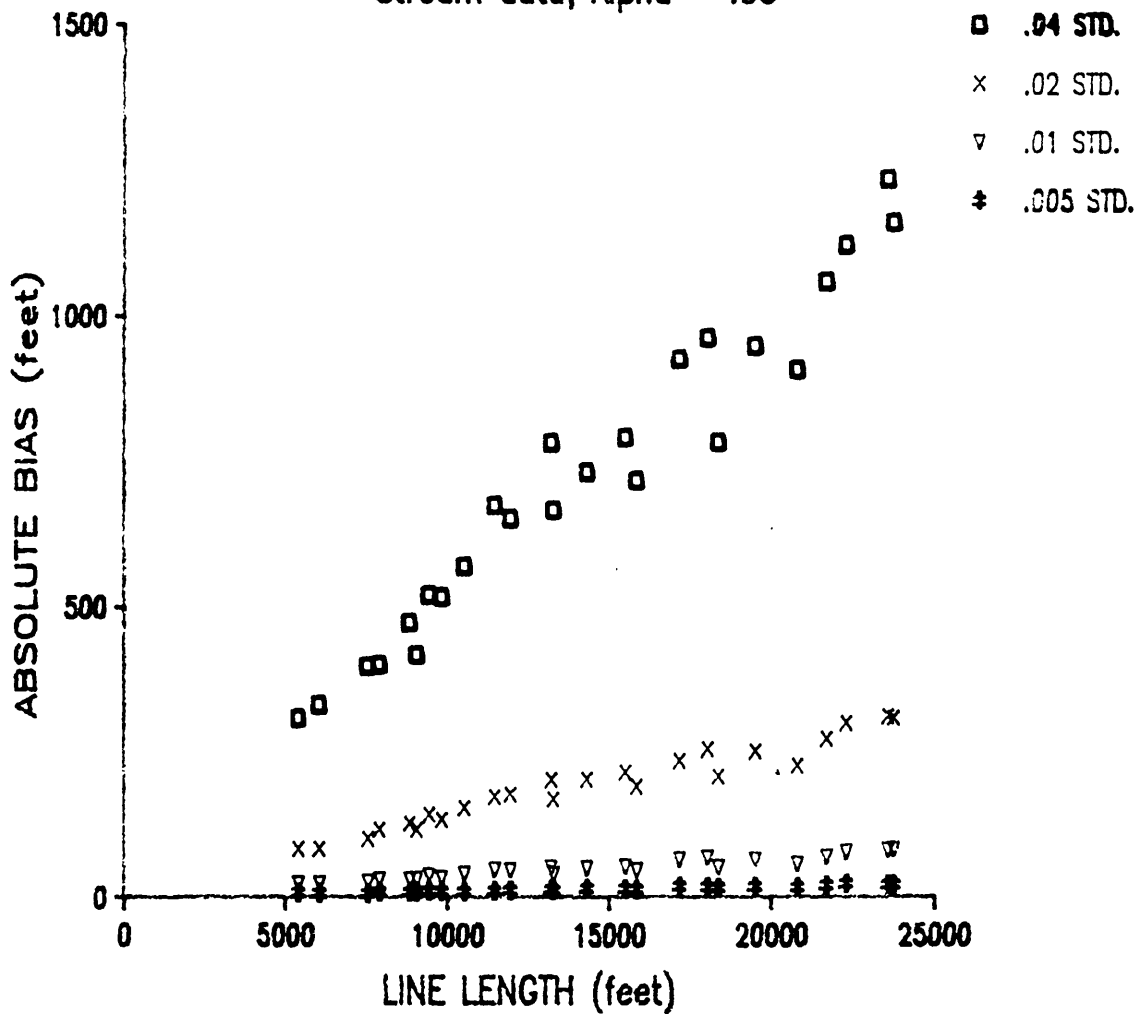
RELATIVE STD DEV VS. POLYGON AREA

Point data, Alpha = .20



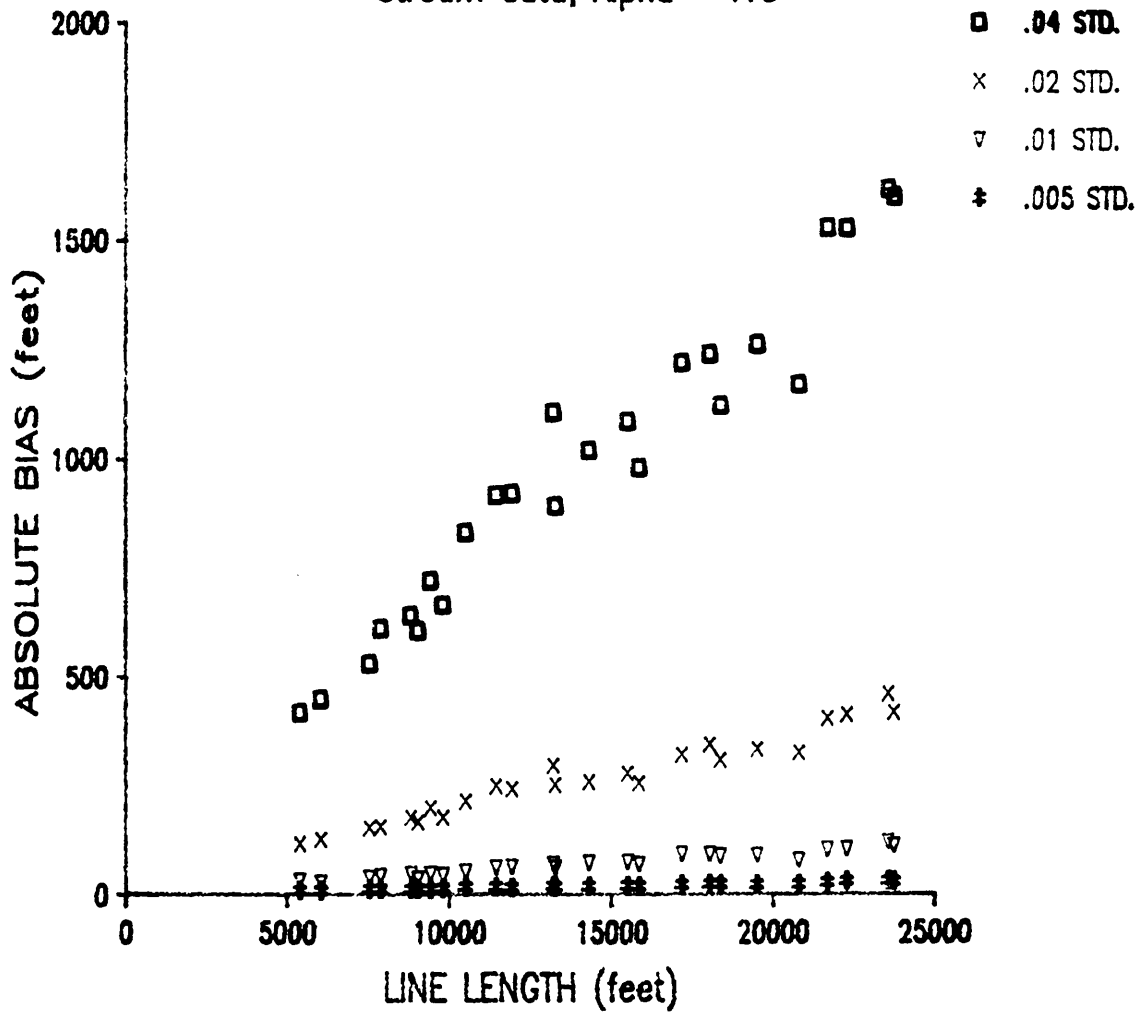
ABSOLUTE BIAS VS. LINE LENGTH

Stream data, Alpha = .05



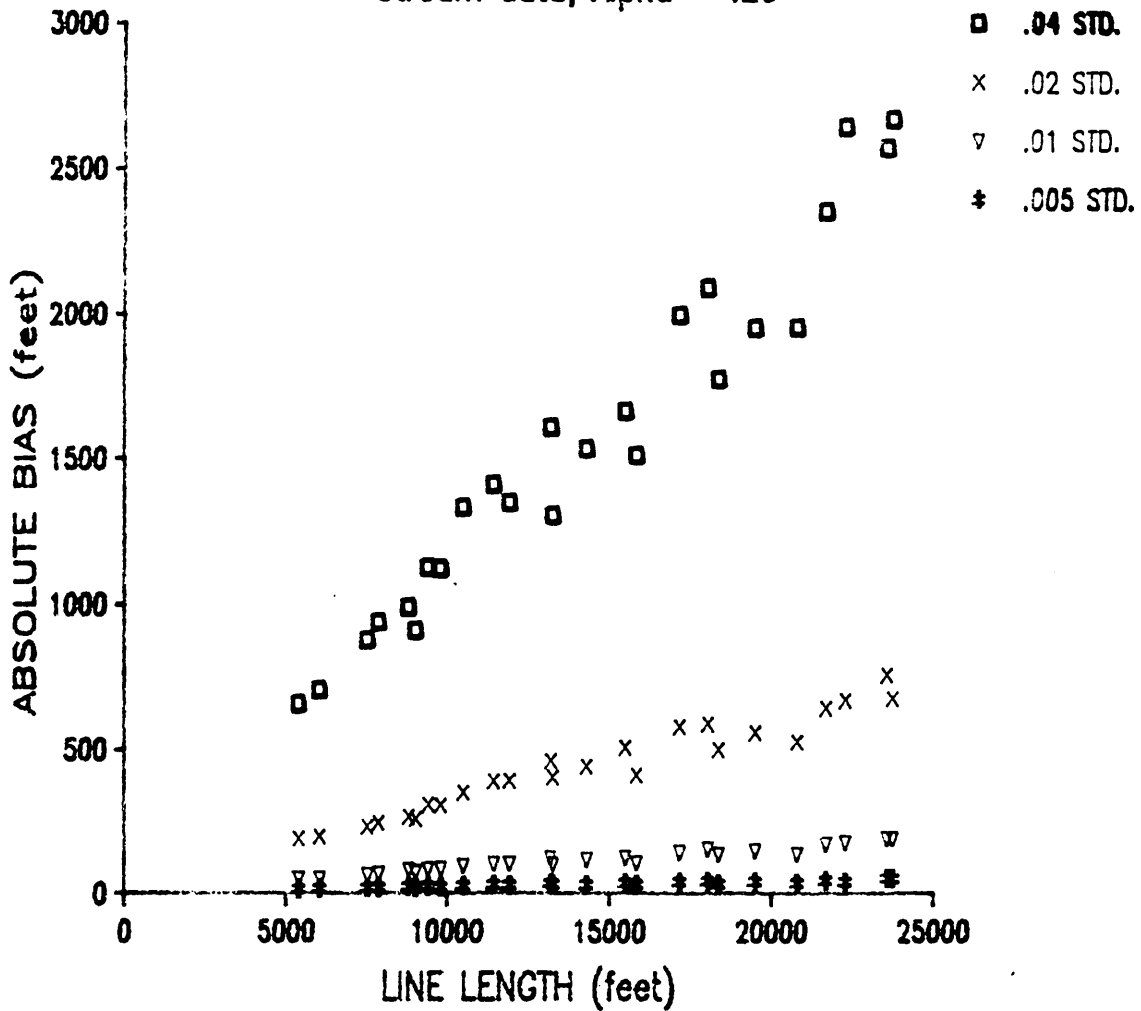
ABSOLUTE BIAS VS. LINE LENGTH

Stream data, Alpha = .10



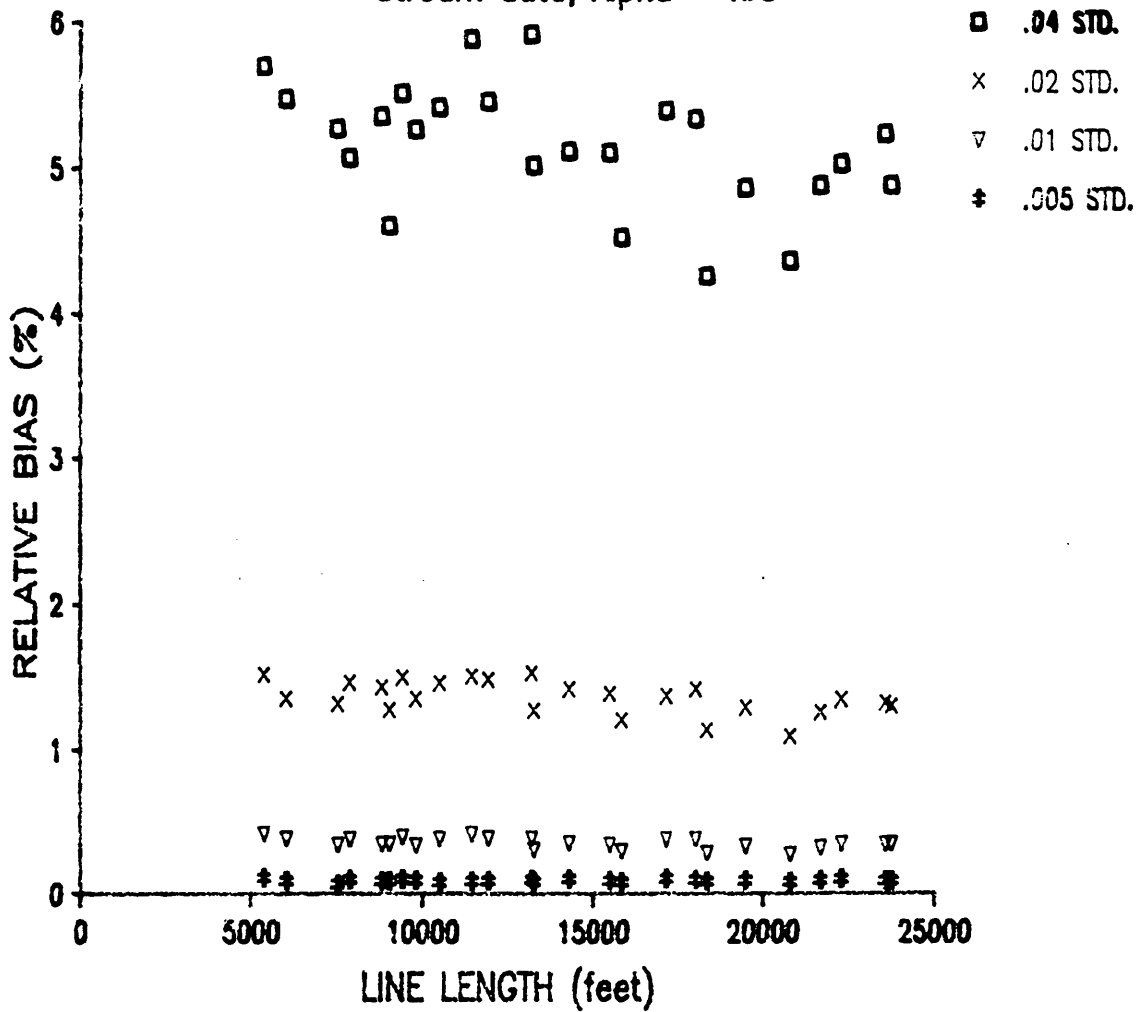
ABSOLUTE BIAS VS. LINE LENGTH

Stream data, Alpha = .20



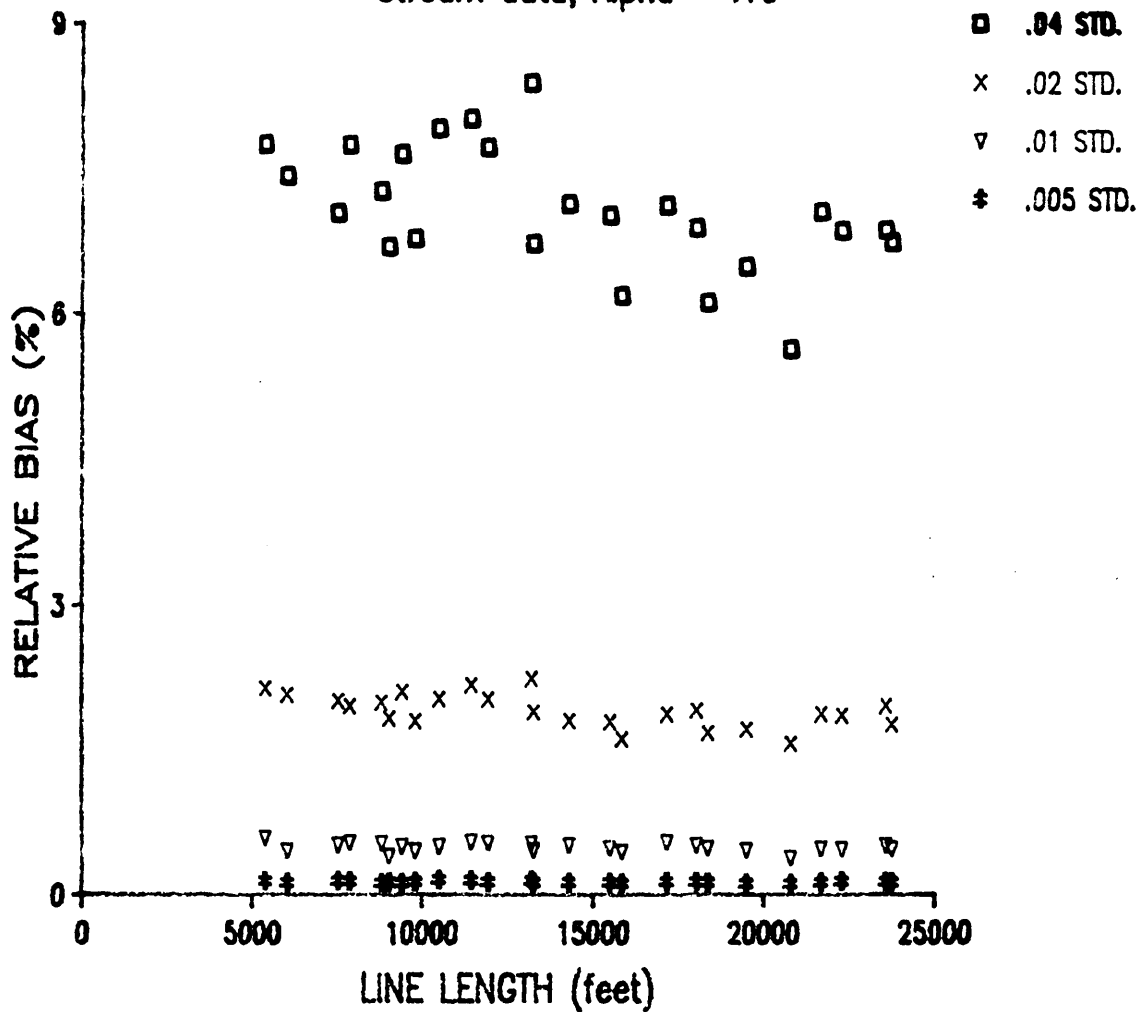
RELATIVE BIAS VS. LINE LENGTH

Stream data, Alpha = .05



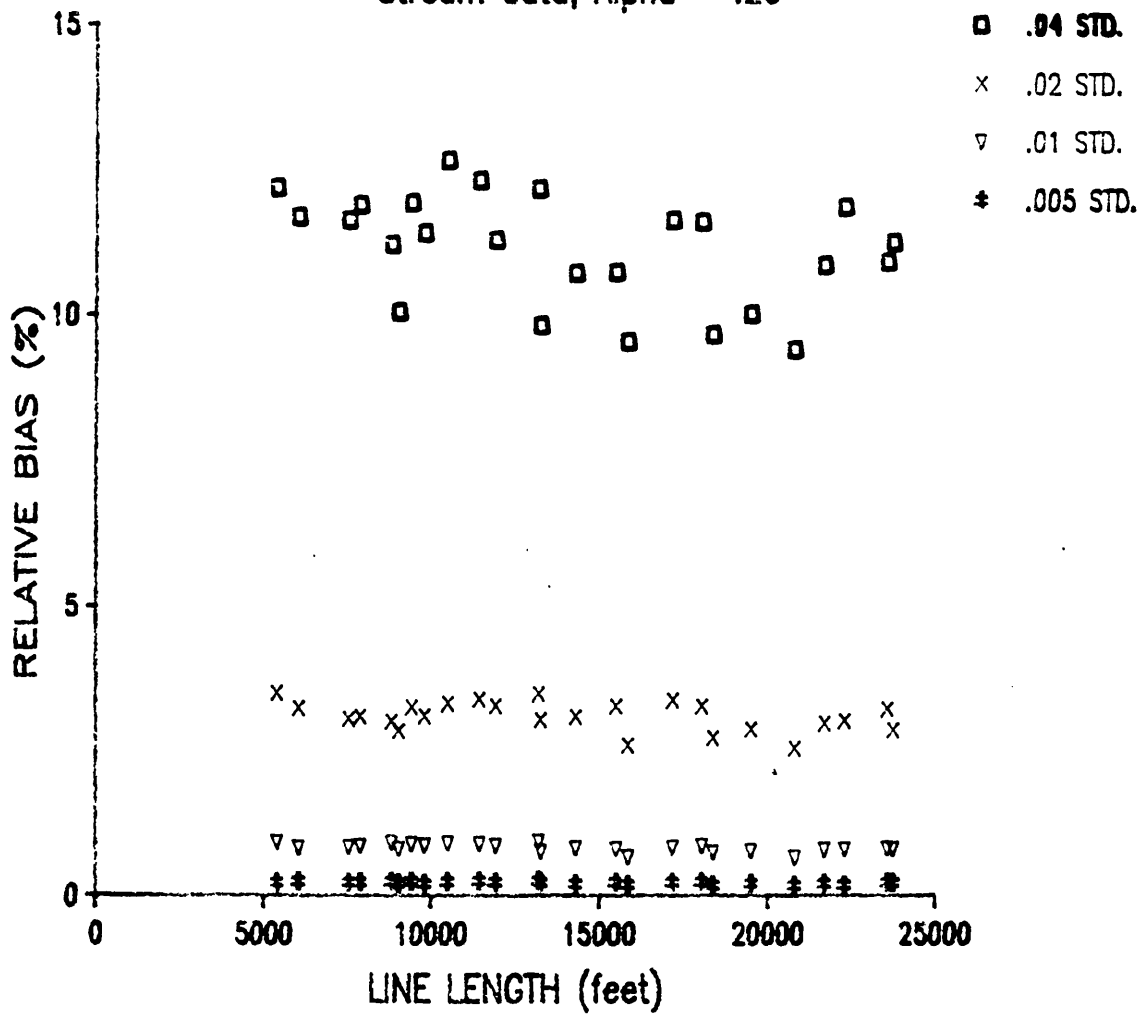
RELATIVE BIAS VS. LINE LENGTH

Stream data, Alpha = .10



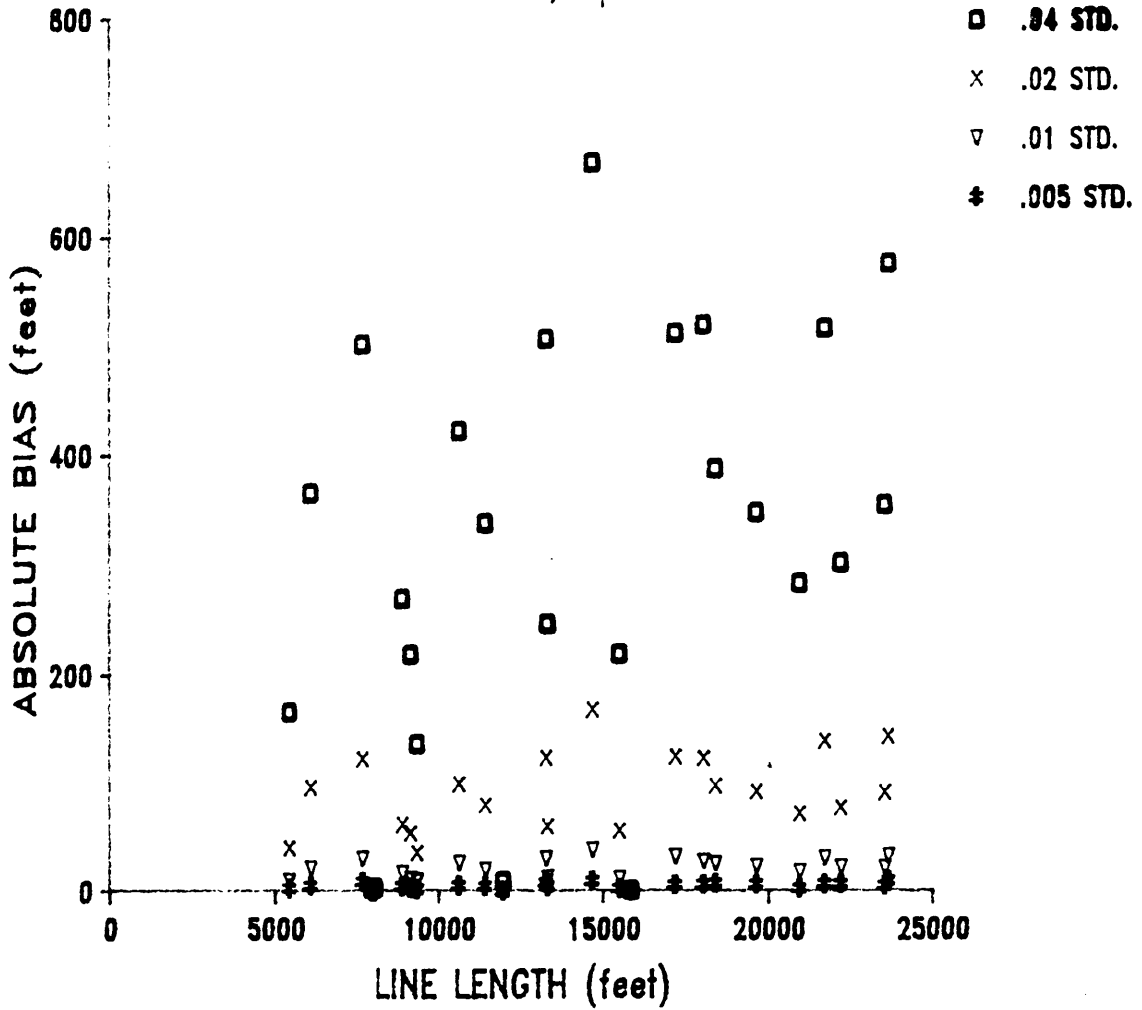
RELATIVE BIAS VS. LINE LENGTH

Stream data, Alpha = .20



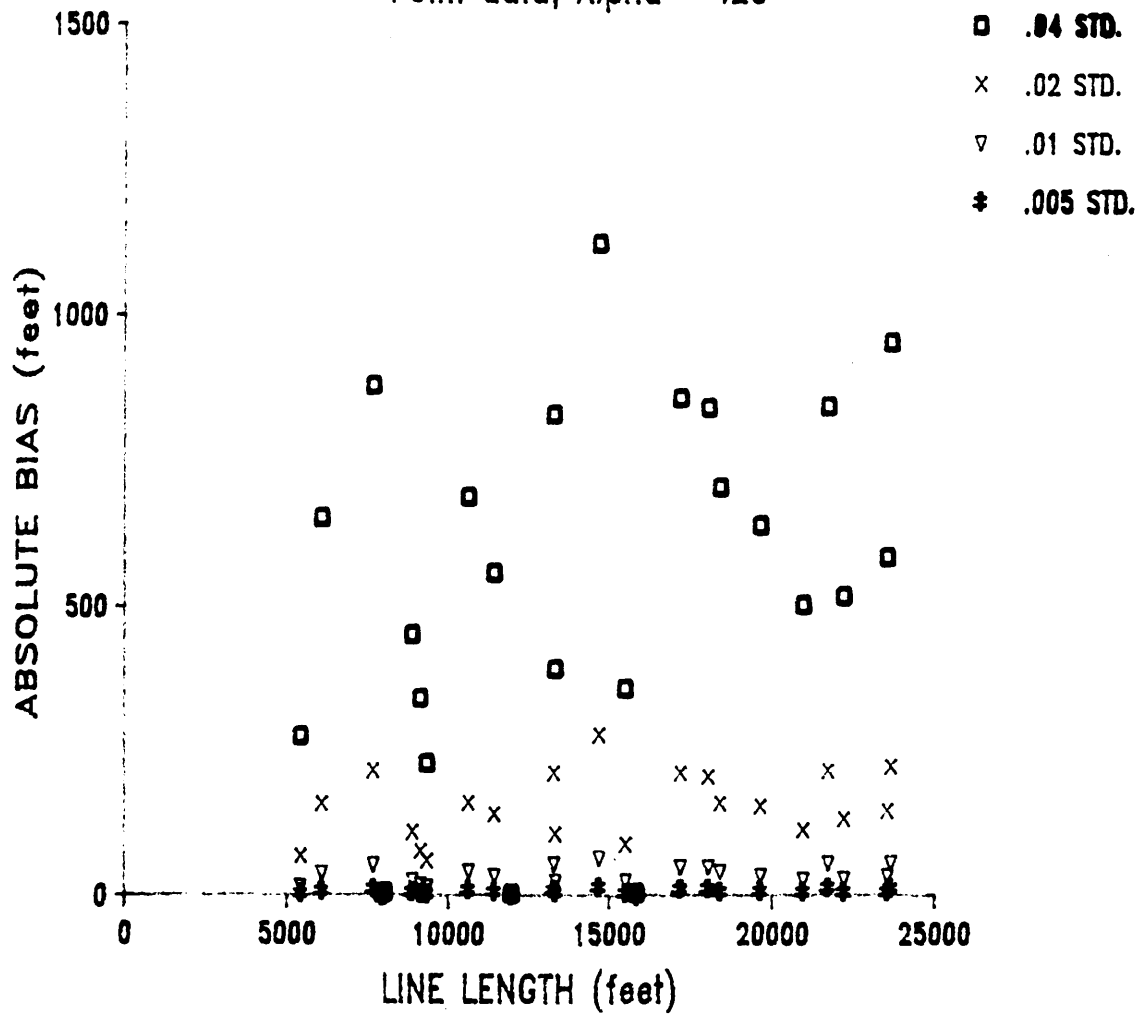
ABSOLUTE BIAS VS. LINE LENGTH

Point data, Alpha = .10



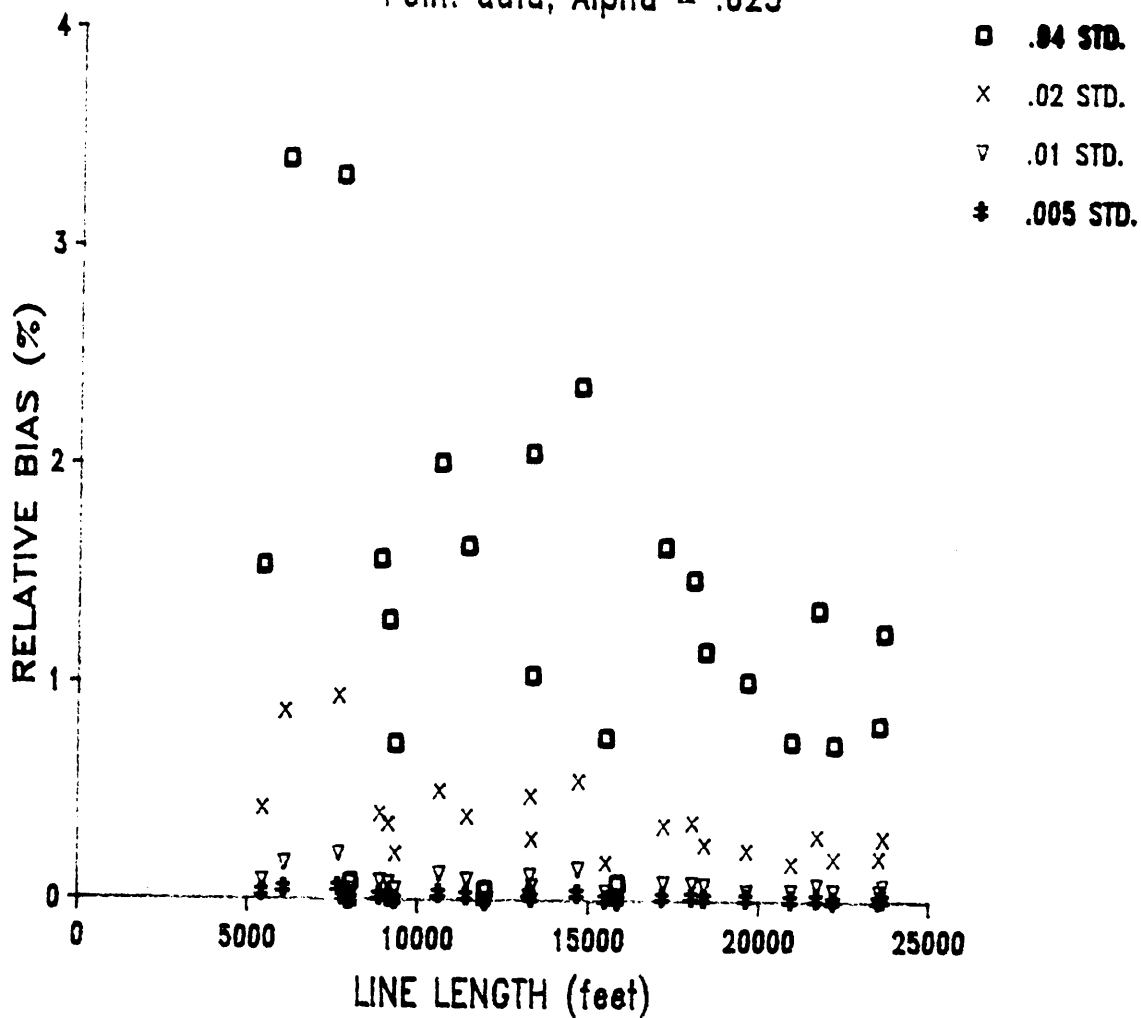
ABSOLUTE BIAS VS. LINE LENGTH

Point data, Alpha = .20



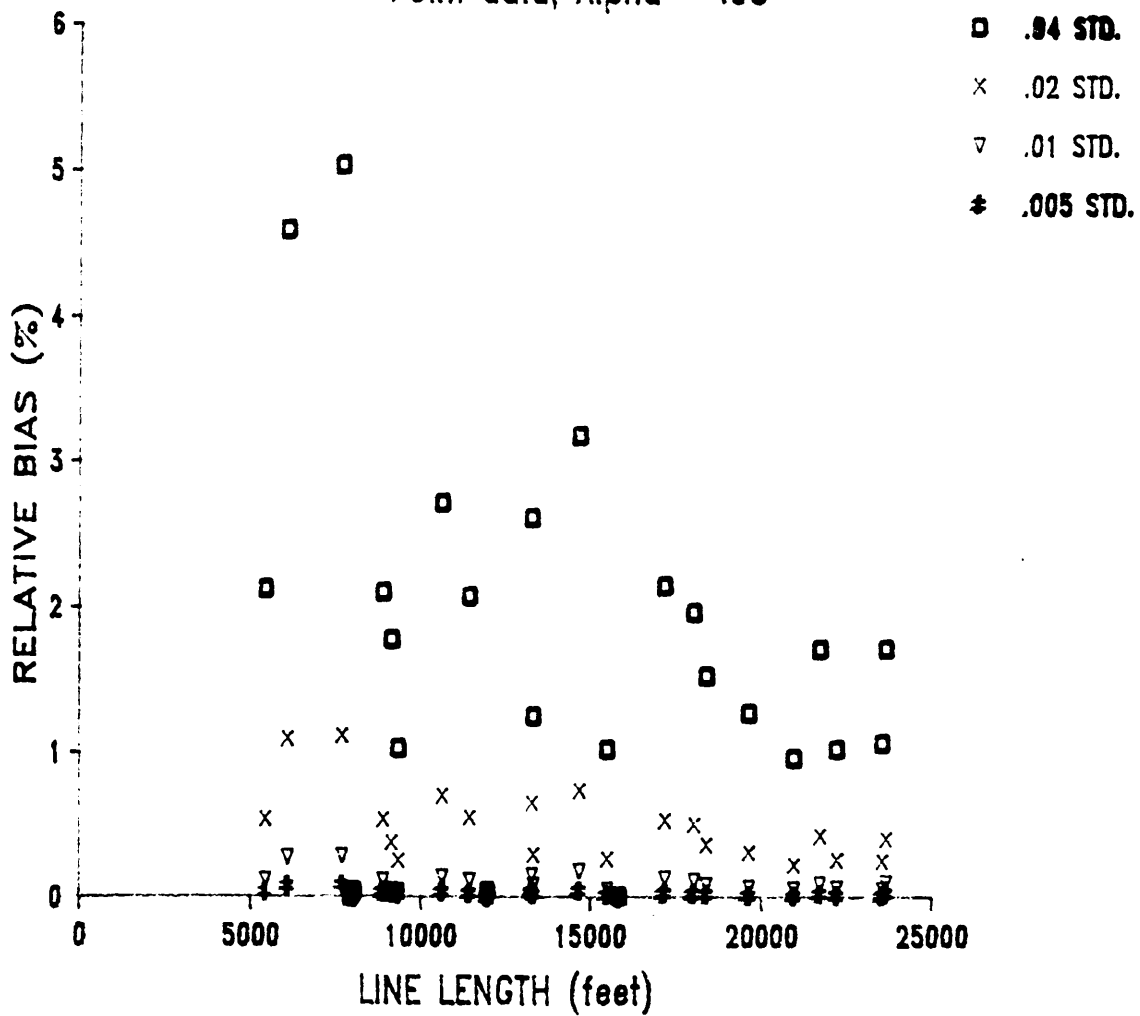
RELATIVE BIAS VS. LINE LENGTH

Point data, Alpha = .025



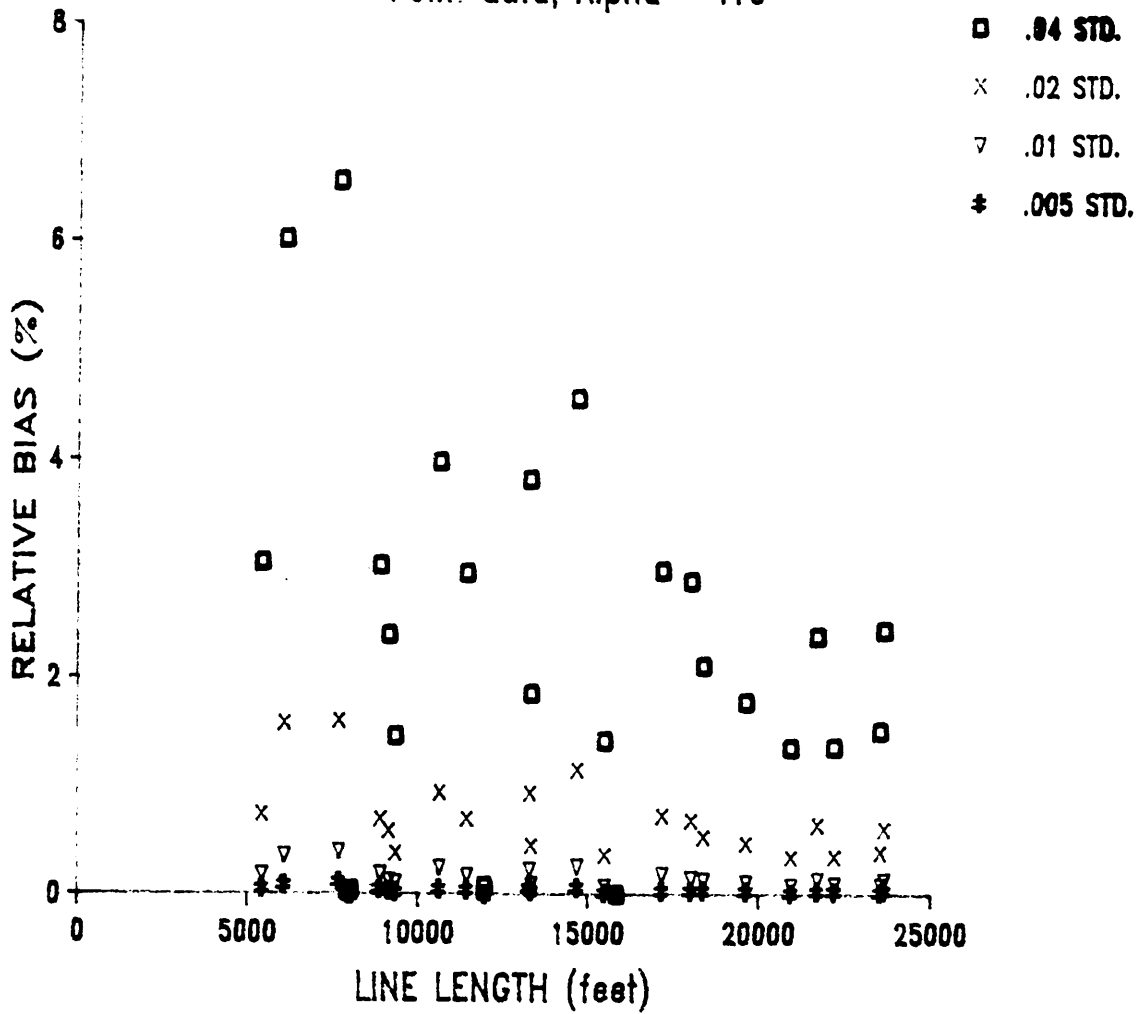
RELATIVE BIAS VS. LINE LENGTH

Point data, Alpha = .05



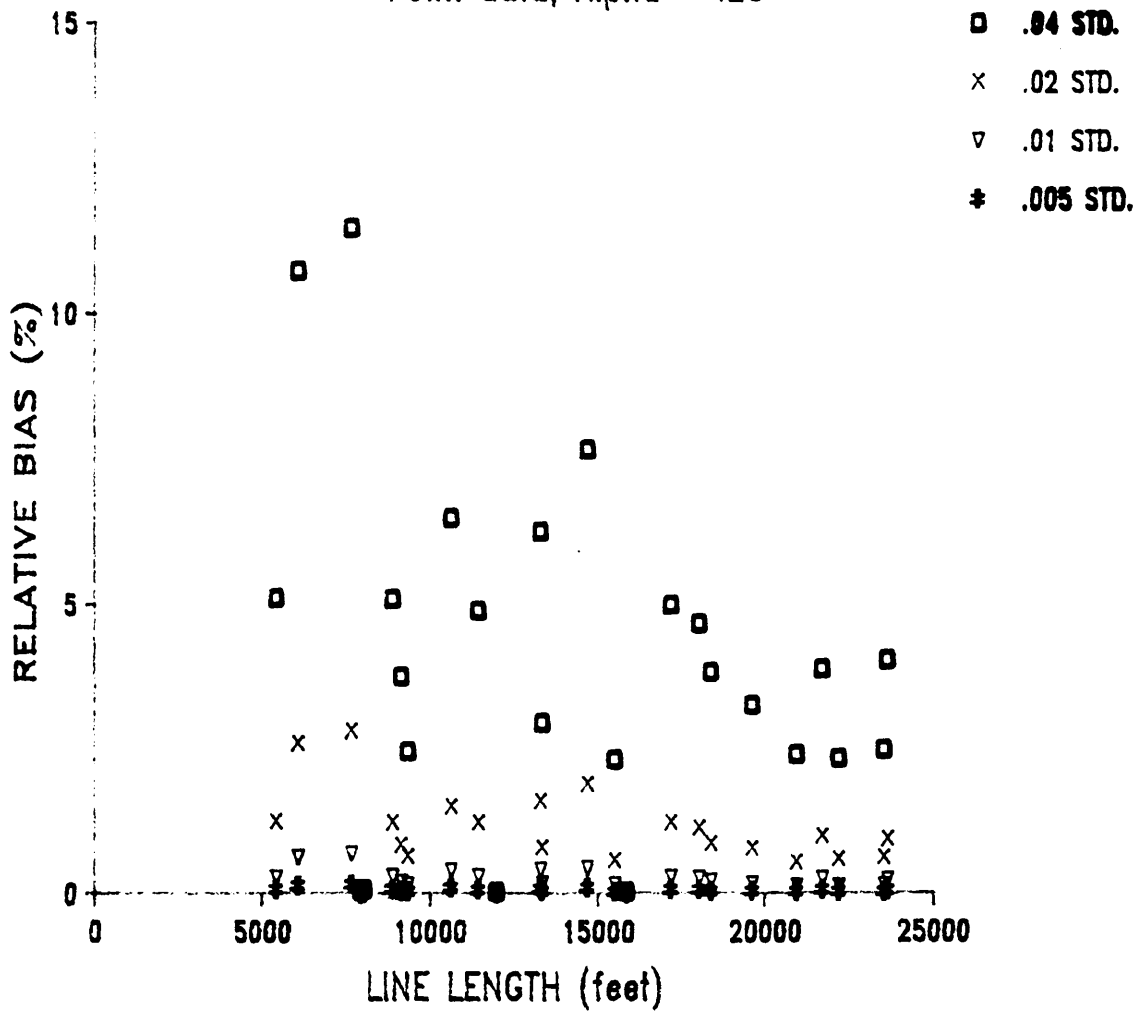
RELATIVE BIAS VS. LINE LENGTH

Point data, Alpha = .10



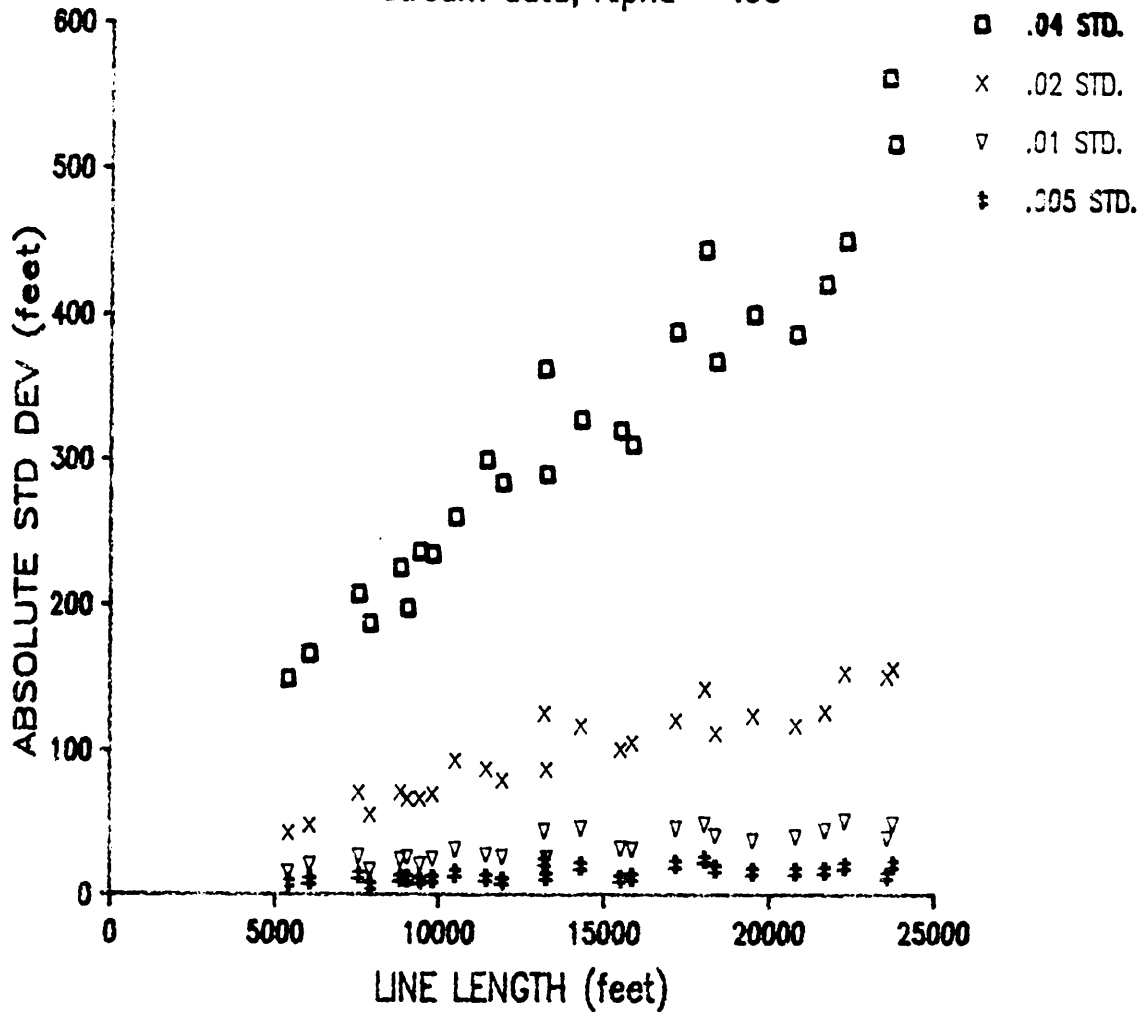
RELATIVE BIAS VS. LINE LENGTH

Point data, Alpha = .20



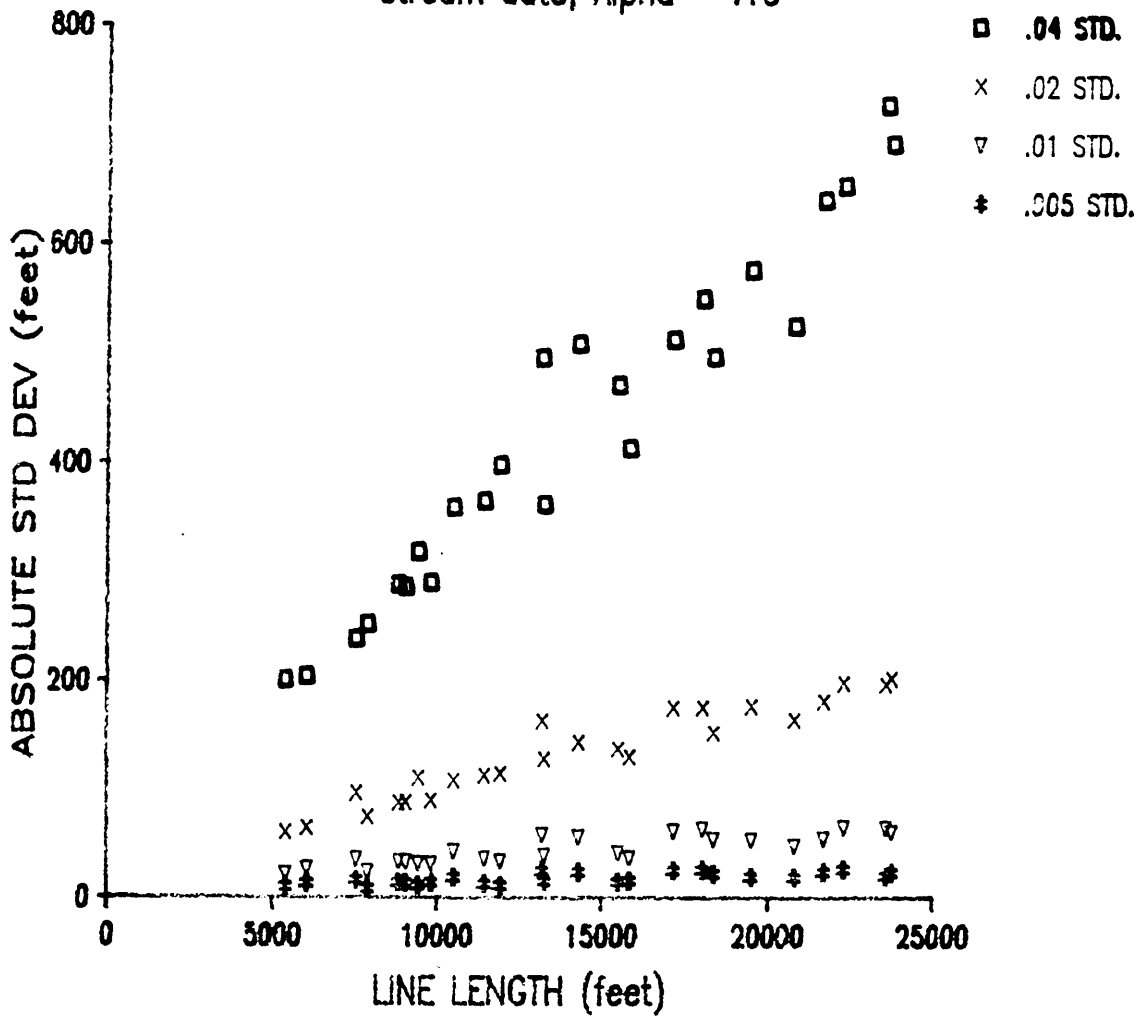
ABSOLUTE STD DEV VS. LINE LENGTH

Stream data, Alpha = .05



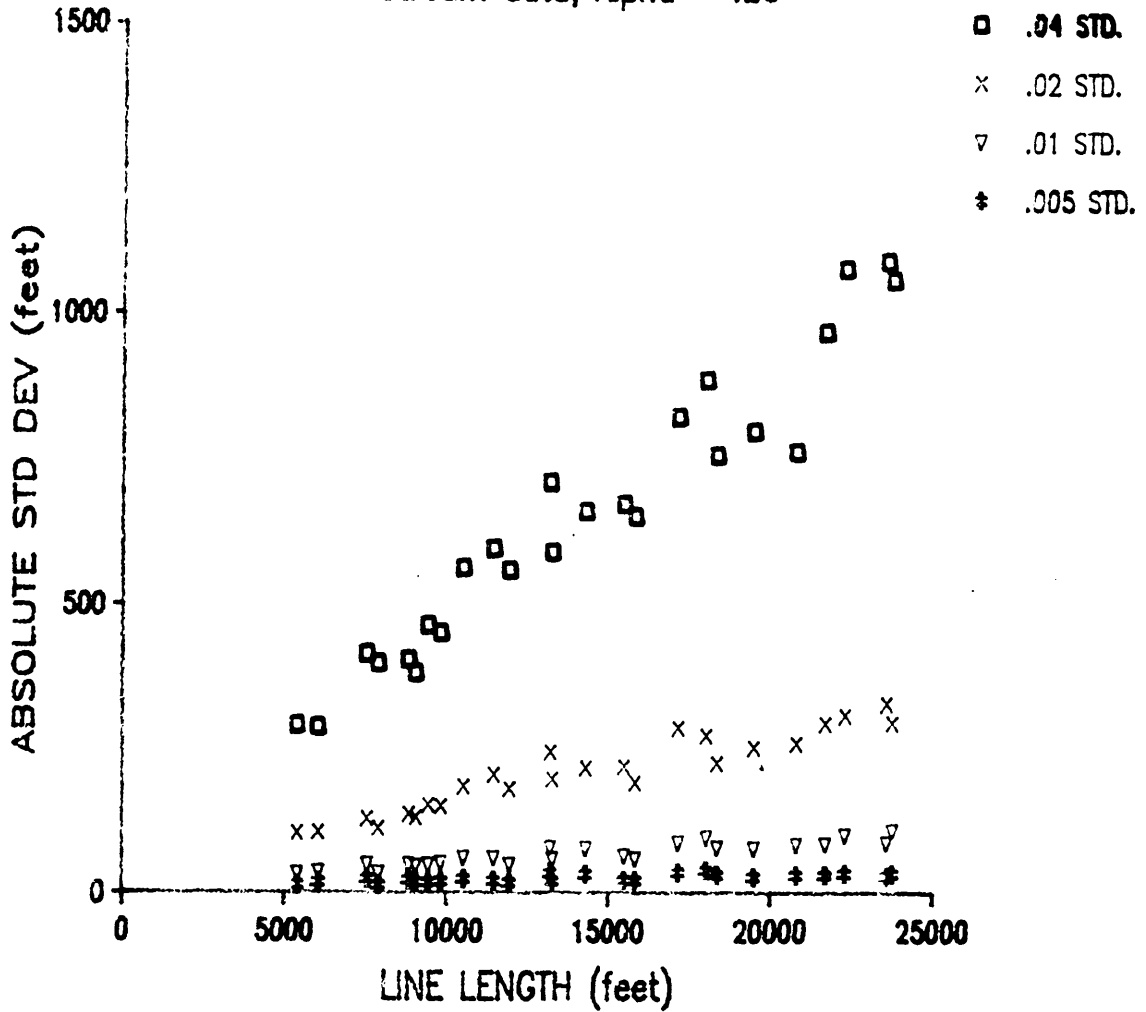
ABSOLUTE STD DEV VS. LINE LENGTH

Stream data, Alpha = .10



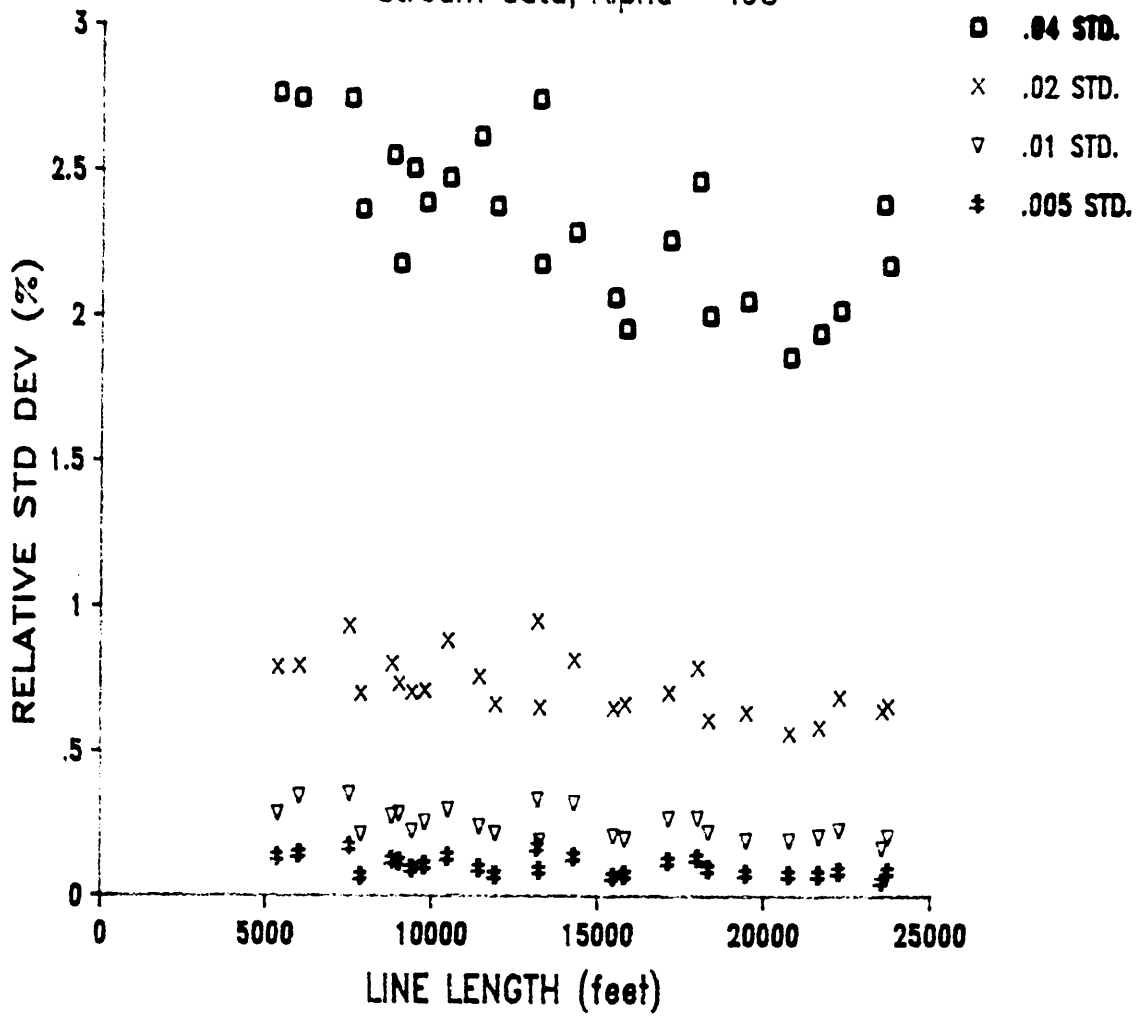
ABSOLUTE STD DEV VS. LINE LENGTH

Stream data, Alpha = .20



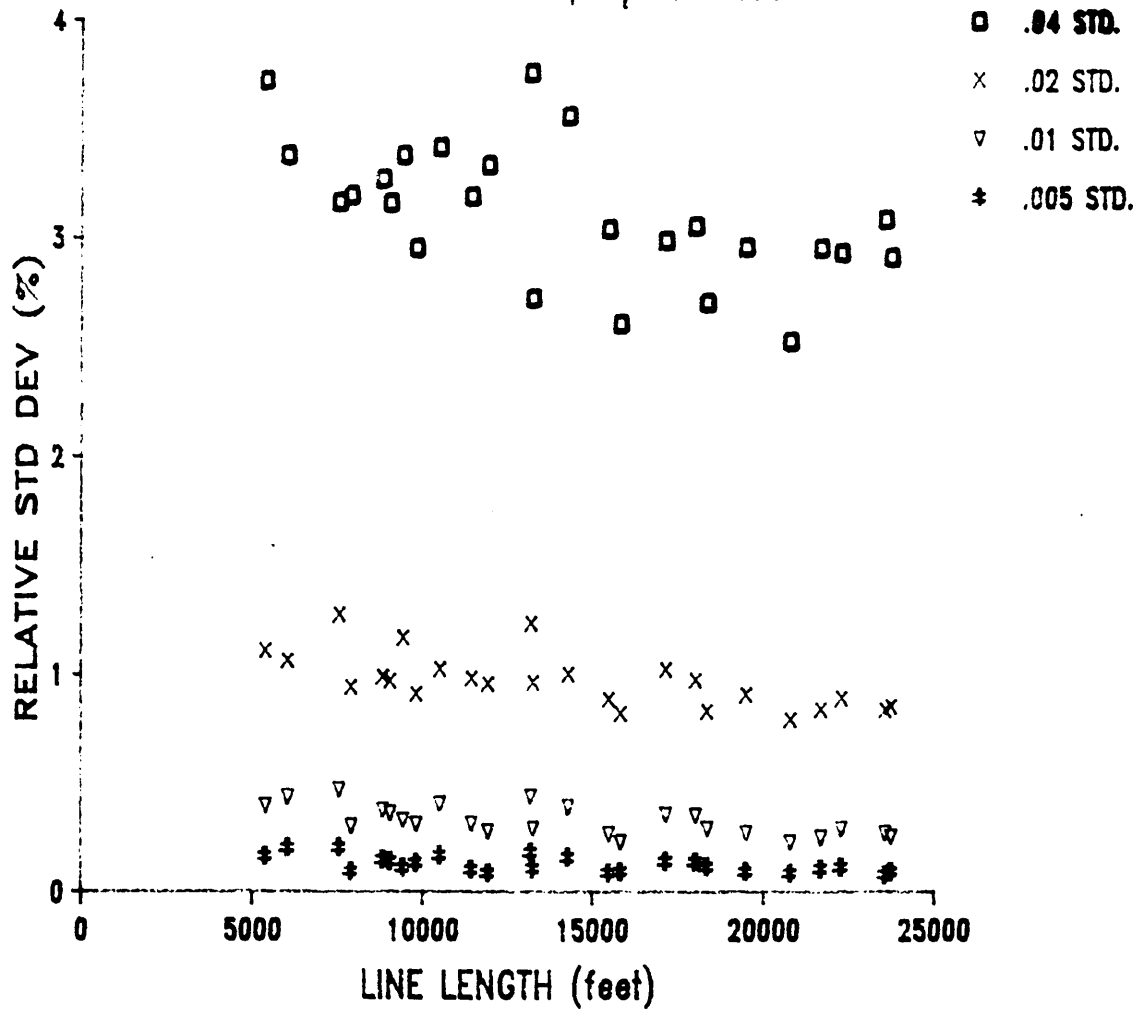
RELATIVE STD DEV VS. LINE LENGTH

Stream data, Alpha = .05



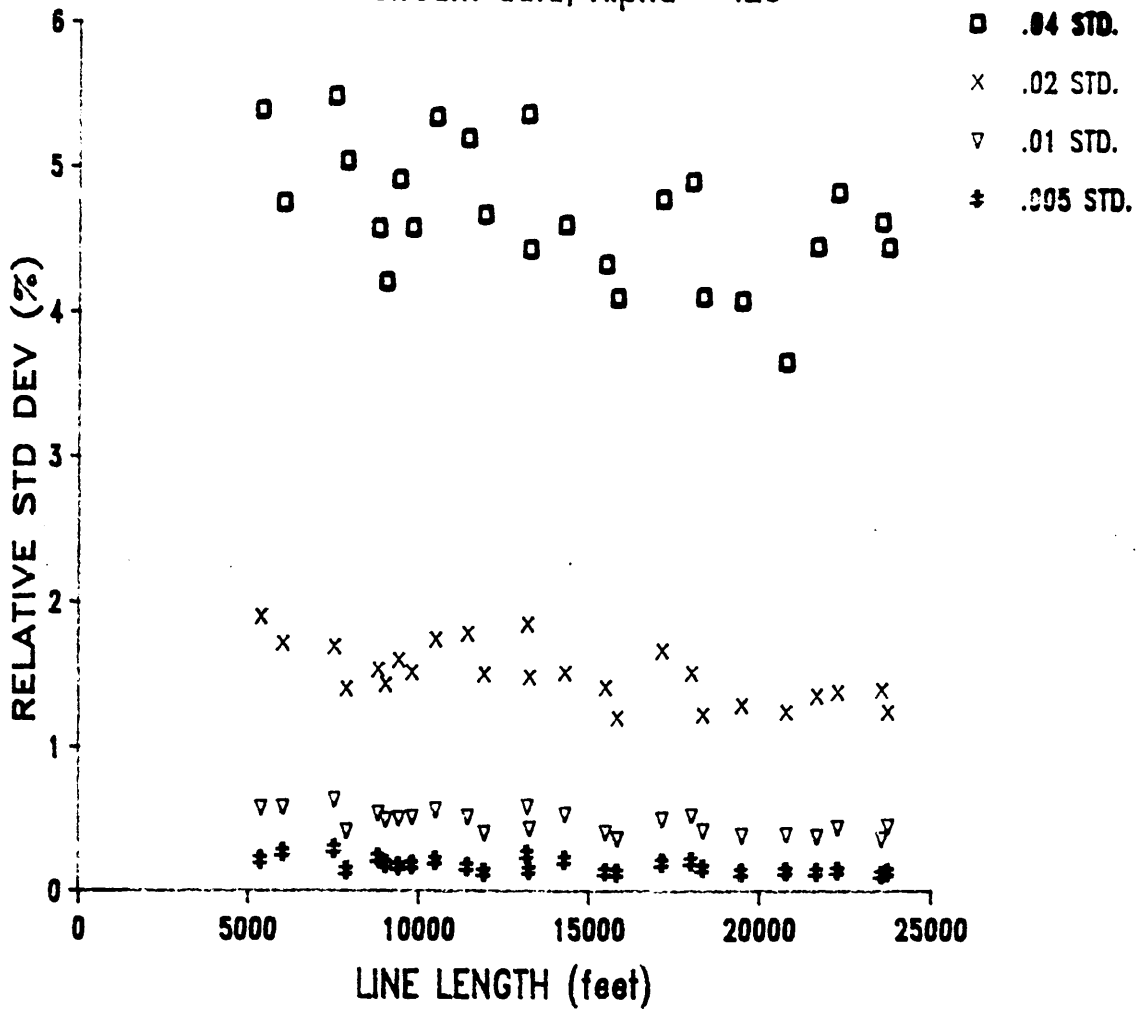
RELATIVE STD DEV VS. LINE LENGTH

Stream data, Alpha = .10



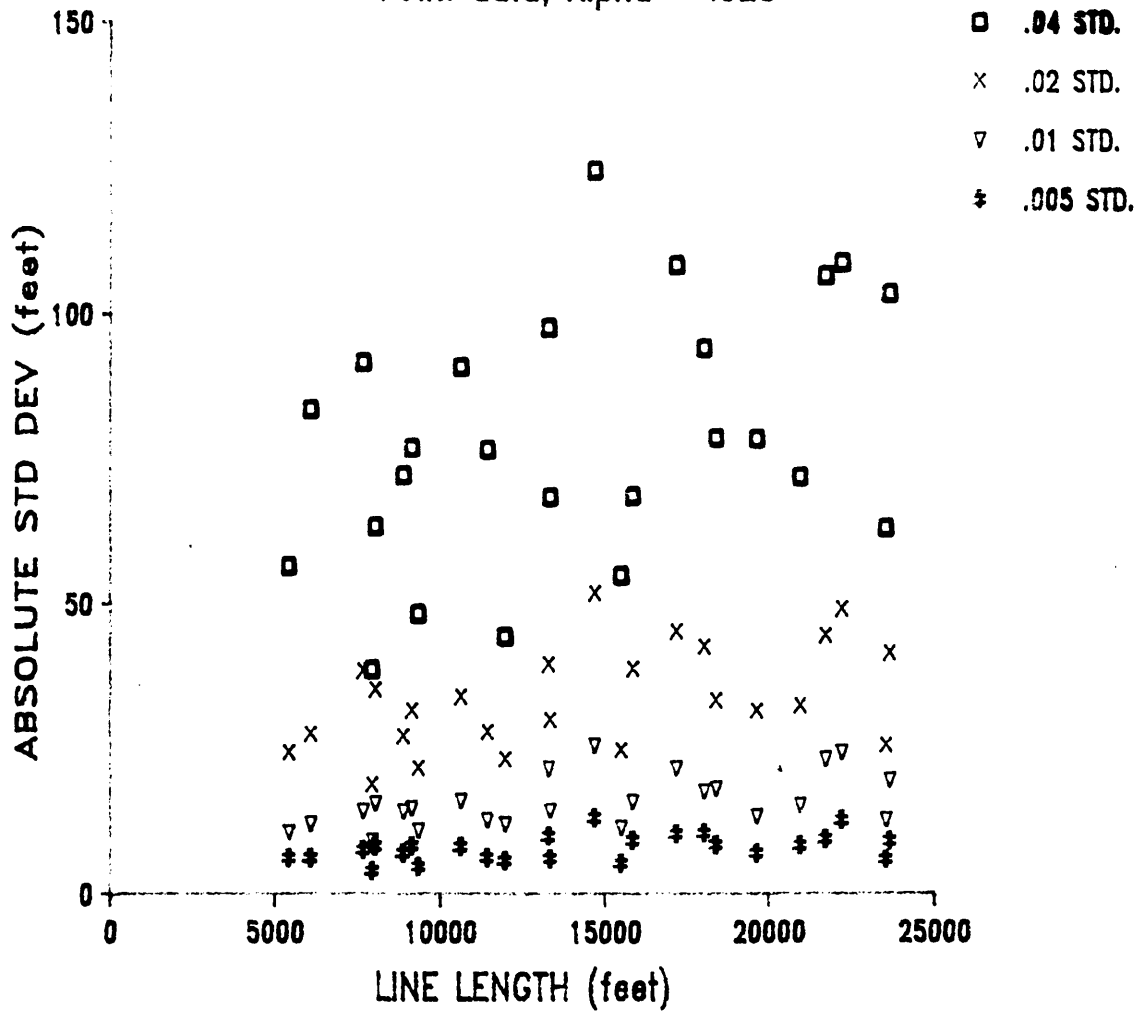
RELATIVE STD DEV VS. LINE LENGTH

Stream data, Alpha = .20



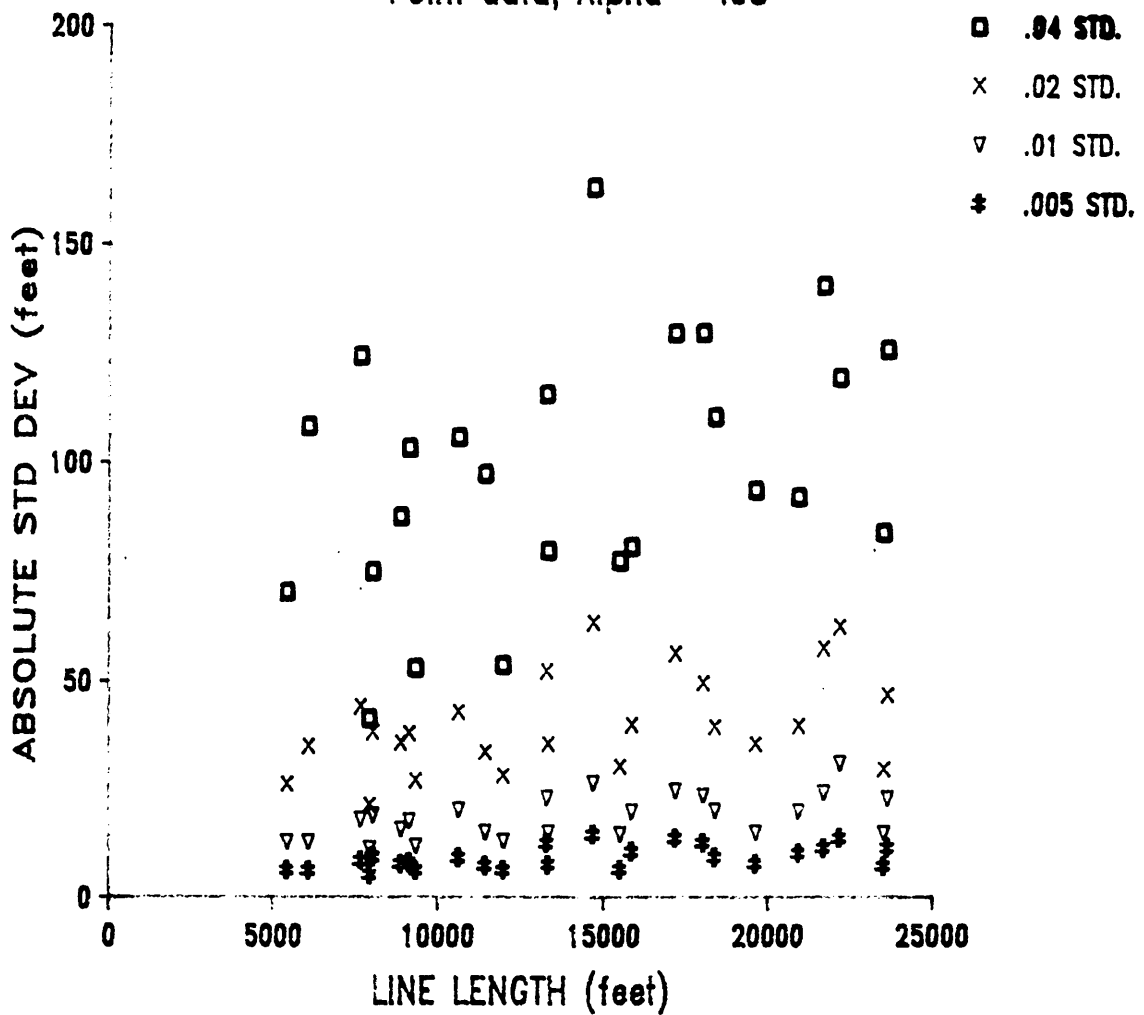
ABSOLUTE STD DEV VS. LINE LENGTH

Point data, Alpha = .025



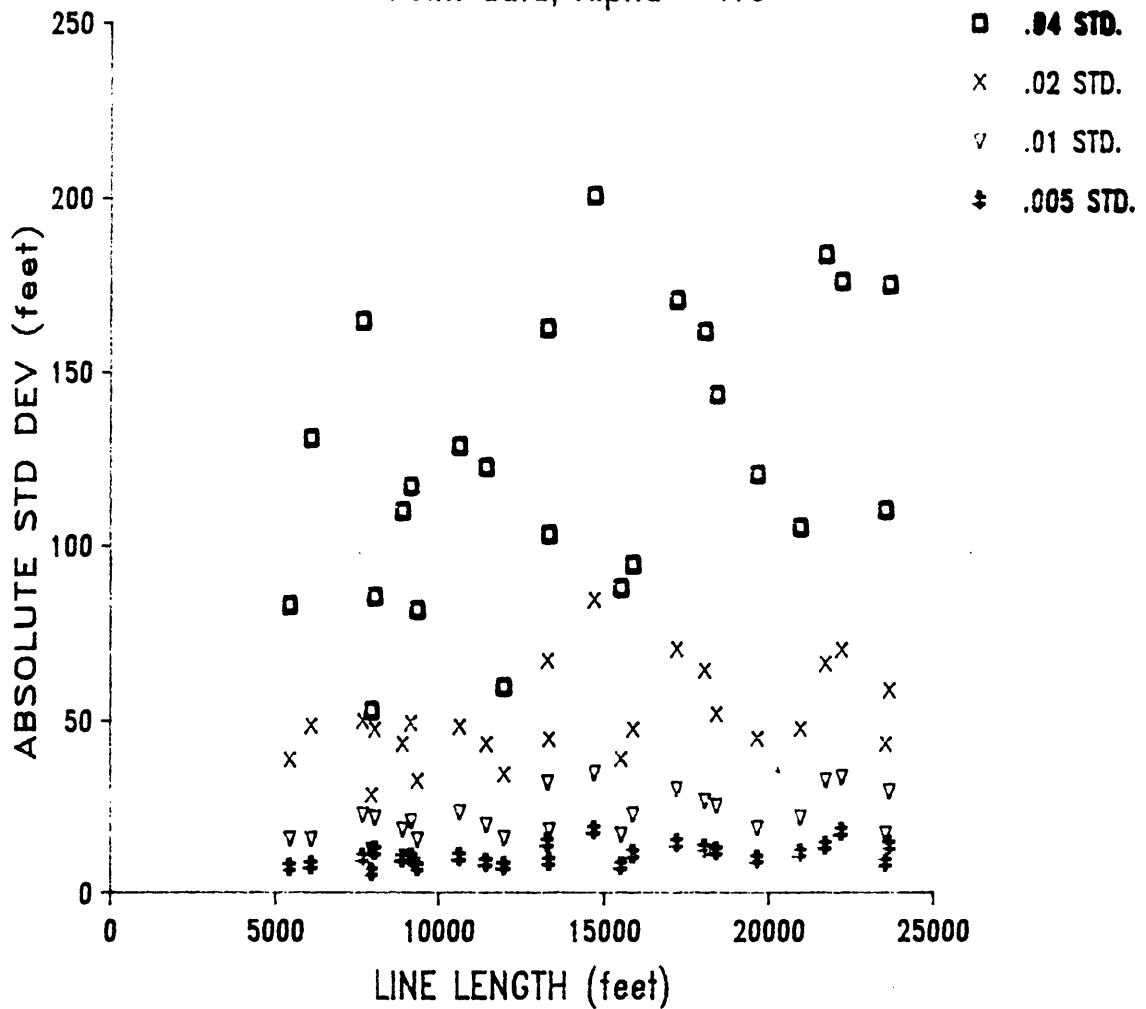
ABSOLUTE STD DEV VS. LINE LENGTH

Point data, Alpha = .05



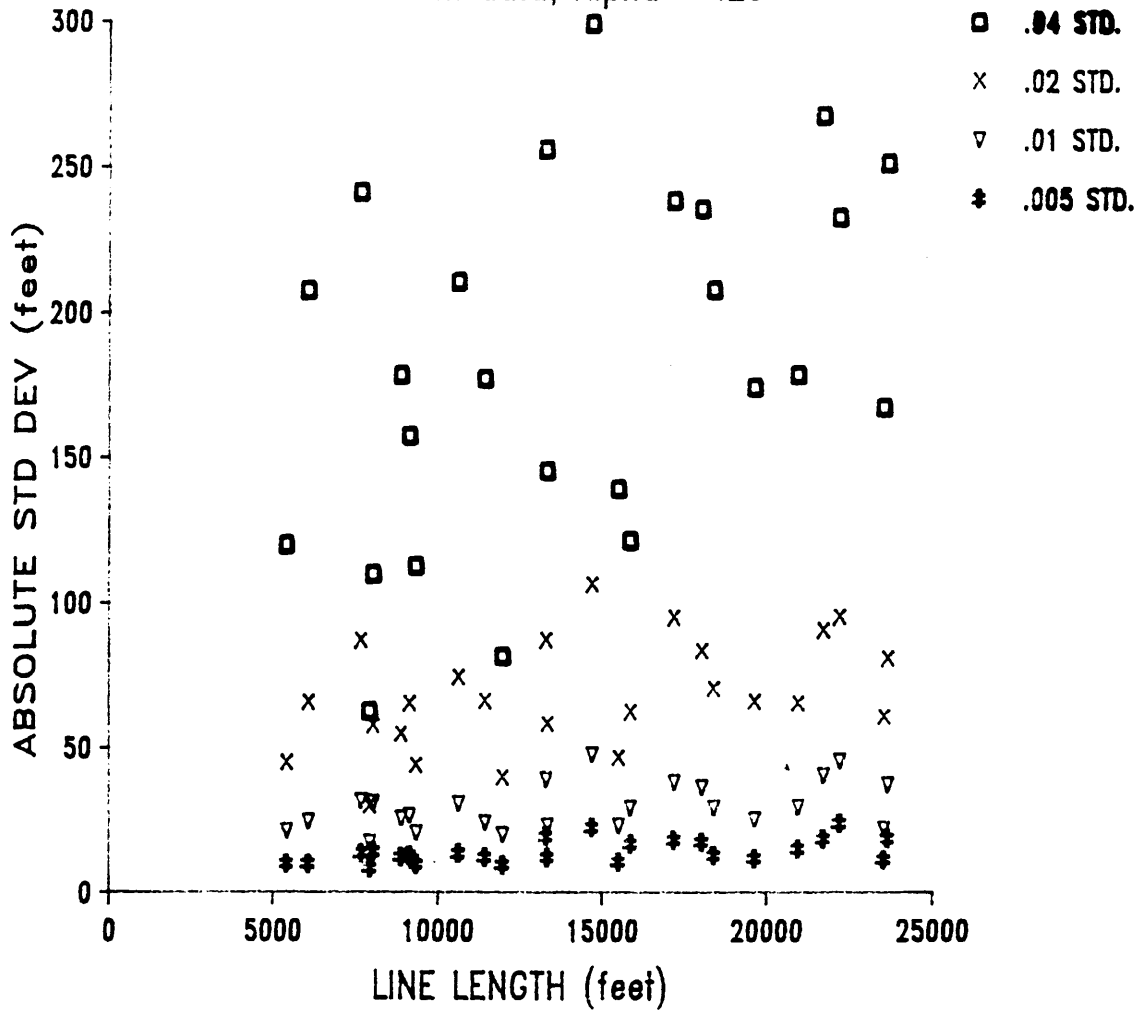
ABSOLUTE STD DEV VS. LINE LENGTH

Point data, Alpha = .10



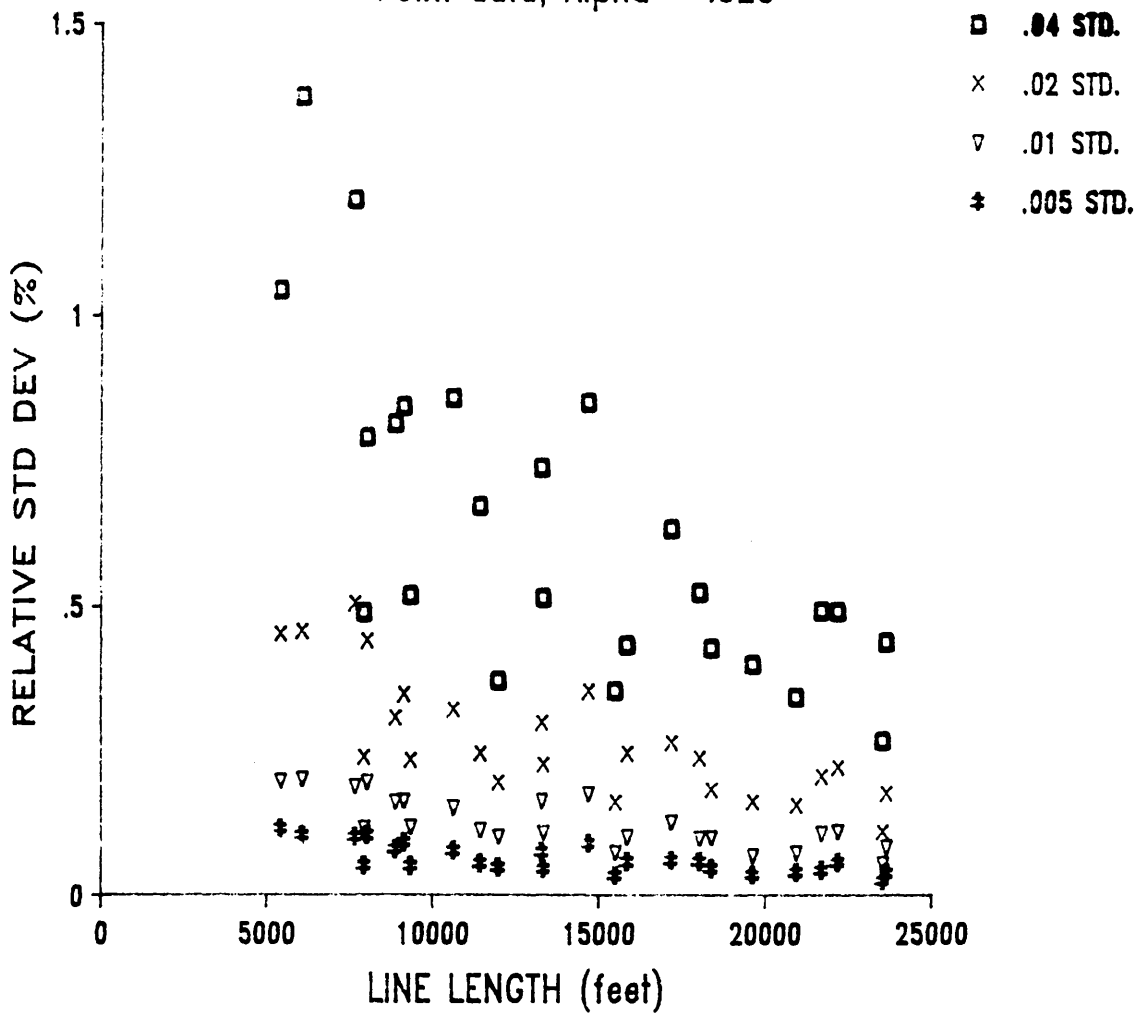
ABSOLUTE STD DEV VS. LINE LENGTH

Point data, Alpha = .20



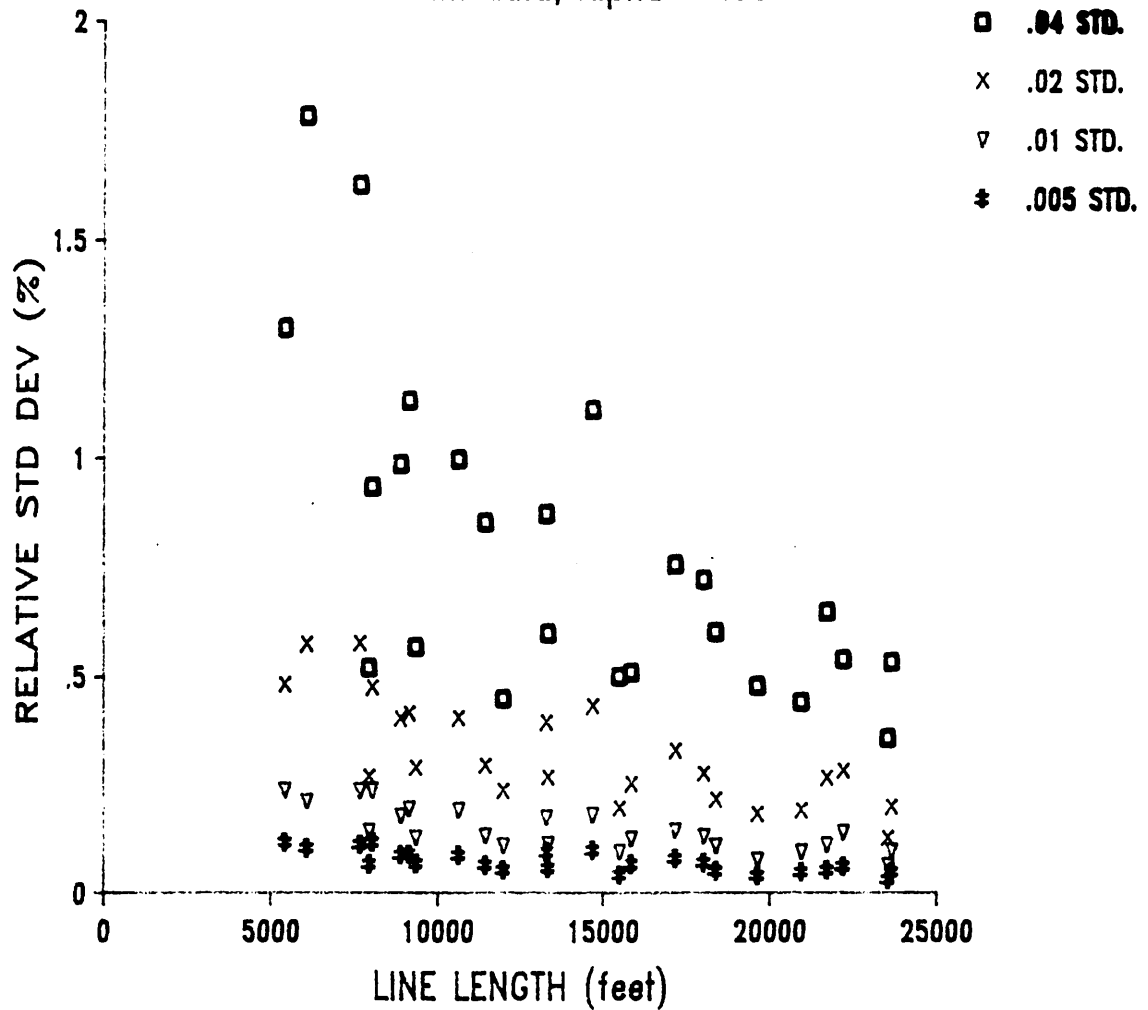
RELATIVE STD DEV VS. LINE LENGTH

Point data, Alpha = .025



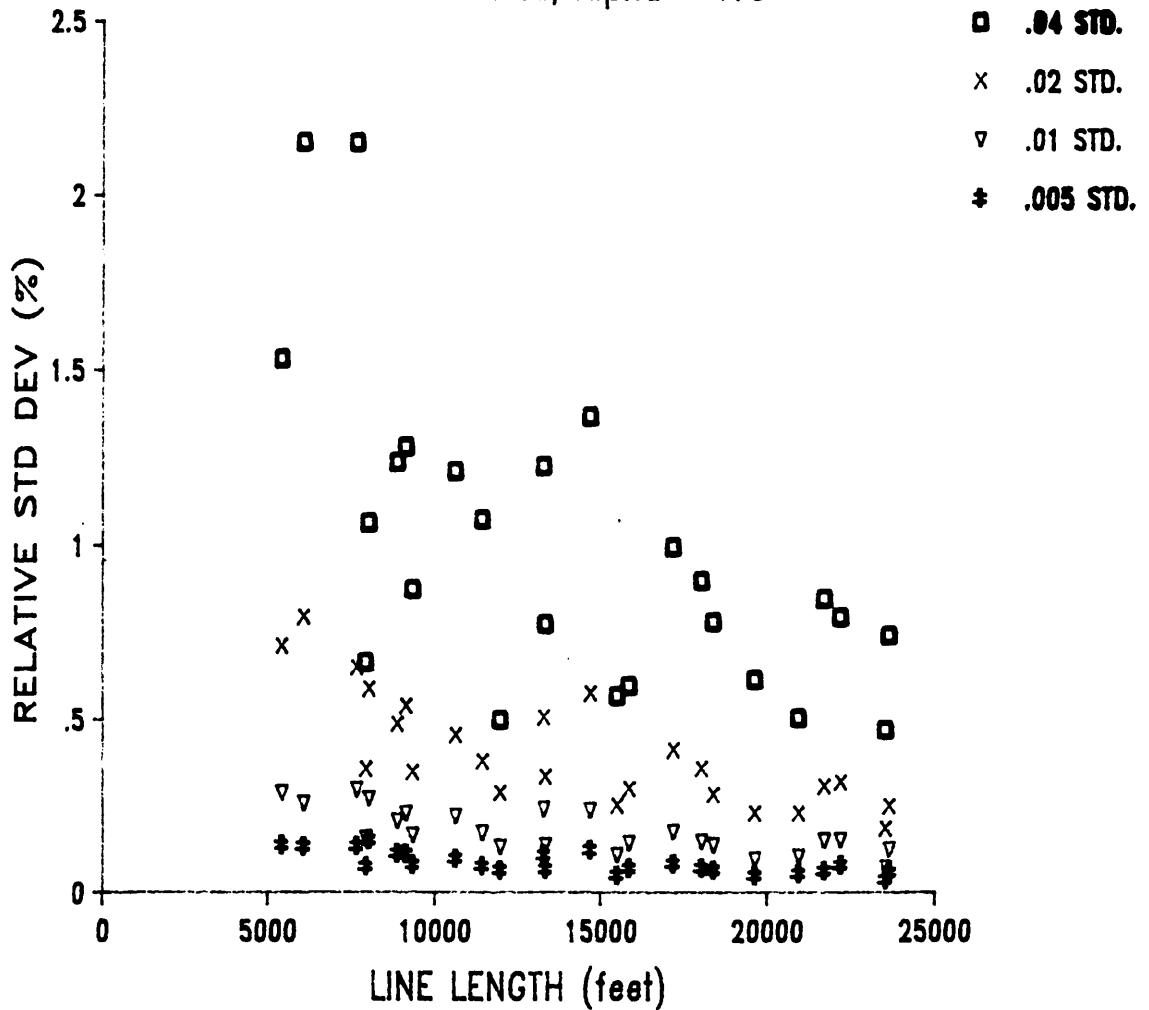
RELATIVE STD DEV VS. LINE LENGTH

Point data, Alpha = .05



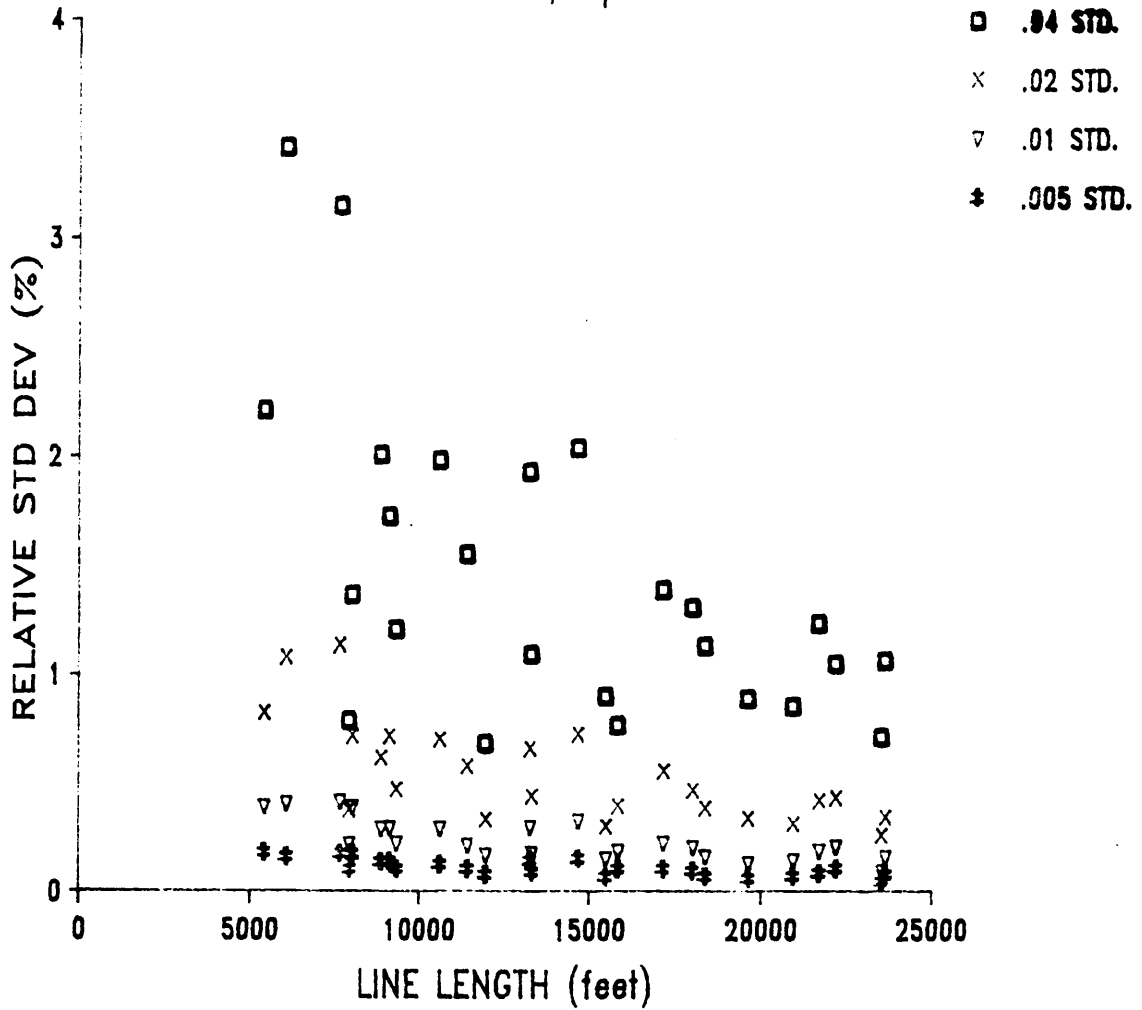
RELATIVE STD DEV VS. LINE LENGTH

Point data, Alpha = .10



RELATIVE STD DEV VS. LINE LENGTH

Point data, Alpha = .20



**The vita has been removed from
the scanned document**