A DATA DENSITY REDUCTION ALGORITHM FOR POST-PROCESSED AIRBORNE LIDAR BATHYMETRIC SURVEY DATA

By

PAMELA S. LAFONTAINE

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2000

This thesis is dedicated to my wonderful husband, Don, and to the Finley Family--my mother, Jo Ann, my father, Chuck, and my brother, Charlie. It is with the love and support of my family that I am able to reach my goals.

ACKNOWLEDGMENTS

I am most grateful to my committee chair, Dr. Bon A. Dewitt, who worked countless hours with me on this project. His open door was always a welcome sight! My other committee members, Dr. Shrestha and Dr. Carter, were also willing to give of their time, and to offer encouragement and guidance.

My sincere appreciation goes to the faculty, staff, and students of the Geomatics Program for the valuable training and support I've received throughout. A special thanks goes to Sandra Russ for her help in keeping everything running so smoothly.

I am indebted to the members of the University of Florida Geoplan Center for the use of their facilities, and for help and encouragement in my programming efforts. Without the expert training of Ilir Bejleri, Avenue would still be a foreign language. I also thank Renee, Balaji, and others who helped me get over some hurdles.

Thanks goes to NIMA for selecting me for this training opportunity and for providing direction in the research aspects of this project. I especially thank Christopher Gede and Dr. Mark Lambrecht for exchanging information and ideas.

Finally, I wish to recognize Jeff Lillycrop, Gary Guenther and the entire SHOALS team for their groundbreaking work in lidar bathymetry. It was their enthusiasm which motivated this study. They eagerly provided me with the data and cooperation I needed to pursue my research.

TABLE OF CONTENTS

page

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

A DATA DENSITY REDUCTION ALGORITHM FOR POST-PROCESSED AIRBORNE LIDAR BATHYMETRIC SURVEY DATA

By

Pamela S. LaFontaine

May 2000

Chairman: Bon A. Dewitt, Ph. D.
Major Department: Civil Engineering

A Data Density Reduction Algorithm (DDRA) for post-processed airborne lidar bathymetric survey data is presented. It may be used to effectively remove "redundant" points from the very large data sets collected by laser systems. A resulting less-dense digital terrain model (DTM) makes the information more manageable for many users. This research includes a review of various data density reduction methods and concepts, and of the history of bathymetric airborne lidar technology. The operational capabilities of the Scanning Hydrographic Operational Airborne Lidar Survey (SHOALS) system are discussed in detail. The algorithm was specifically designed for use on lidar data collected by this U. S. Army Corps of Engineers' platform. It should, however, perform equally well on any terrain data expressed in x, y, and z coordinates.

The capabilities of the Data Density Reduction Algorithm are identified and compared to the measurement error in the original surveyed data. With the choice of appropriate input parameters, the algorithm can effectively thin data while staying within the RMS threshold set by the vertical error. In an effort to establish its lower limit of effectiveness, the DDRA is also compared to both systematic and random rarefaction processes. Nearly every test case was favorable in that the algorithm proved more effective than the other two thinning techniques. In general, the root mean square (RMS) of the thinning process increases with the percentage of data removed. This accuracy assessment value is thus related to the two user input parameters of radius and tolerance. Once the original data points are thinned beyond a given margin, the algorithm becomes less capable.

A Fourier analysis technique for assisting in the choice of the radius parameter is also tested and discussed. Surfaces are generated from the lidar data points of three surveyed test sites. Each surface is then processed using a Fast Fourier Transform, and a magnitude image is produced. From this image, a profile is taken across the frequency domain to indicate the magnitude of signal in each frequency component, or harmonic. These profiles depict a peaked signal curve fading into noise. By identifying the harmonic range in which the signal is concentrated, a required sampling frequency can be computed. It is concluded that this sampling frequency would serve as an ideal radius parameter in the Data Density Reduction Algorithm. Unfortunately, the precision of the technique is restricted by possible error in the interpretation of the Fourier magnitude profile to determine the harmonic range of the signal.

INTRODUCTION

Various users of a particular data set do not necessarily have the same application in mind. They also may not have access to computer systems of equal capability. For this reason, it is sometimes necessary to modify a data set so that it is appropriate for the user and the task.

This thesis presents an algorithm that is designed to intelligently reduce the density of a digital terrain model. The algorithm is incorporated into a computer program written in the Avenue object-oriented programming language (See Appendix A). The program was tested on lidar data using an ArcView Geographic Information System (GIS), as discussed in the Results and Discussion section.

Although this data reduction technique may be applicable for any digital terrain model, it has been specifically developed for the U.S. Army Corps of Engineers (USACE) who manage the Scanning Hydrographic Operational Airborne Lidar Survey (SHOALS) system. SHOALS is an airborne bathymetry system which uses a helicopter, or fixed-wing, mounted laser to collect soundings in shallow waters of up to about 50 meters in depth. At 200 laser pulses per second, an altitude of approximately 400 meters above the water's surface, and a speed of 30 meters per second (60 knots), SHOALS surveys at a rate of approximately 8 square kilometers per hour (3 square miles per hour). The hundreds of thousands of depth measurements provided are generally separated by no more than about 4 meters (13 feet). This results in an enormous amount of data being collected (Lillycrop, Irish, and Parson 1997).

SHOALS surveys are so dense that some USACE customers have difficulty utilizing and managing the data. Therefore, the operators of the system require a means of effectively thinning their surveys for these user groups. They are searching for an "intelligent" method which keeps more points in areas of high variability, and uses fewer points where the terrain is relatively unchanging. In other words, a process of decimation which simply removes every tenth point, regardless of its significance, is unacceptable.

Furthermore, it is very important to the SHOALS team that the positional accuracy of the data produced by the system be retained. To ensure data integrity, they want to avoid “binning” or gridding. Although these methods are frequently used to reduce large data sets, they tend to generalize the data. Gridding establishes new latitude (y) and longitude (x) locations that are centered in data cells. The z-value of each grid cell is often a nearest-neighbor estimate of the actual depths surrounding that position. The USACE prefers depths to be mapped in their surveyed locations, and does not want to risk losing accuracy by using techniques that involve interpolation.

Once a data density reduction (DDR) method is designed, it is important to evaluate just how successfully it retains the information content of the original data. How well does a terrain surface generated from the complete set of raw data points compare to a surface based only on the selected points in a rarefied data set? To determine this, a statistical evaluation technique has been incorporated into the main program.

This study was conducted under the constraint of maintaining positional accuracy of the original survey data. With that underlying goal, the answers to three research questions were pursued:

1. What types of parameters affect intelligent data density reduction?

2. What are reasonable values for these parameters?

3. How well does a reduced DTM represent the surveyed terrain data?

This thesis is divided into five main sections. The first provides background on the operations and history of airborne lidar, with an emphasis on the SHOALS system. Next there is a review of data density reduction concepts with various methods being contrasted and compared. The third section (Materials and Methods) presents a DDR algorithm and a procedure used to evaluate its success. It includes a description of two statistical hypothesis tests with which the method was analyzed—the Fisher test, or F-test, and the chi-square test. The Fourier analysis technique investigated as a means of isolating the most appropriate input parameters is also reviewed. In the fourth section, the results of statistical comparisons of the newly developed DDR algorithm with systematic and random thinning approaches are discussed. More significantly, the results of the algorithm as statistically compared to the stated vertical measurement error of the data are presented. Results from the Fourier analysis used in an attempt to improve the effectiveness of the algorithm are included as well. The final Summary and Conclusions section offers answers to the three research questions posed earlier as well as recommendations for further development of concepts presented throughout this thesis.

CONCEPT AND HISTORY OF AIRBORNE LIDAR BATHYMETRY

What is Airborne Lidar Bathymetry (ALB)?

Just as radar is an acronym for radio detection and ranging, lidar is an acronym for light detection and ranging. An airborne lidar bathymeter (ALB) is an instrument which uses light to calculate water depths. Since the light is generally produced by a laser (light amplification by stimulated emission of radiation), the word laser is often interchanged with the word lidar. An ALB may be defined as either an airborne laser bathymeter or an airborne lidar bathymeter (Guenther, Thomas, and LaRocque 1996). The device is carried under, or within, an aircraft (helicopter or fixed-wing). It emits pulses of light which travel from the transmitter to the water's surface below. At the air/water interface, some of the light energy is reflected, while the rest passes into the water column. The water column has a complex effect on light pulses. In the end, the light may be either absorbed or reflected by the seabed, or by particles or obstructions in the water itself (Lillycrop, Irish, and Parson 1997).

It is the time difference, between the light reflected back to the receiver from the water's surface and the light reflected off the sea bottom, that is used to determine the water's depth. However, there are many factors that complicate this method. For instance, the light pulses are generally directed toward the water at an angle off-nadir and fired in a swath pattern which zigzags across the flight path. This configuration allows for optimum collection and survey efficiency, but requires that all angles be carefully accounted for in the mathematical calculations used to compute time and distance. There are also environmental considerations such as the speed of light versus salinity, the amount of background light (noise) present, and the effects of surface waves. These various biases must be detected and removed from the signals received in order to generate accurate depth measurements (Guenther, Eisler, Riley, and Perez 1996).

Not all wavelengths of electromagnetic energy are suitable for water penetration. In fact, only a very small window of green light is used for this task. Airborne lidar or laser bathymetry (ALB)[1], uses a scanning pulsed green laser transmitter to transmit light to the water's surface and beyond. (Other wavelengths may be used in conjunction with the green, but not for bottom detection.) A telescope, light detectors, amplifiers and analog-to-digital converters receive and process the signals which are reflected back to the system. Digital waveforms are stored for post-flight precise calculations of depth, while approximated depths are generated in flight for quality control purposes (Guenther, Eisler, Riley, and Perez 1996).

In a typical ALB system, aircraft altitude is about 200-500 meters, scanner nadir angle is 15-20 degrees forward of nadir, and the swath stretches 150-250 meters across the flight path. The scan pattern can vary, and is controlled by a pivoting, or rotating, mirror which directs the laser beam. Light pulses that penetrate the water's surface, scatter and expand into a cone whose interior angle and cross-section increase with depth (See Figure 1). Maximum penetration depth is most strongly influenced by water clarity. It can range from over 50 meters

---

[1] ALB is also frequently referred to as airborne lidar or laser hydrography (ALH).

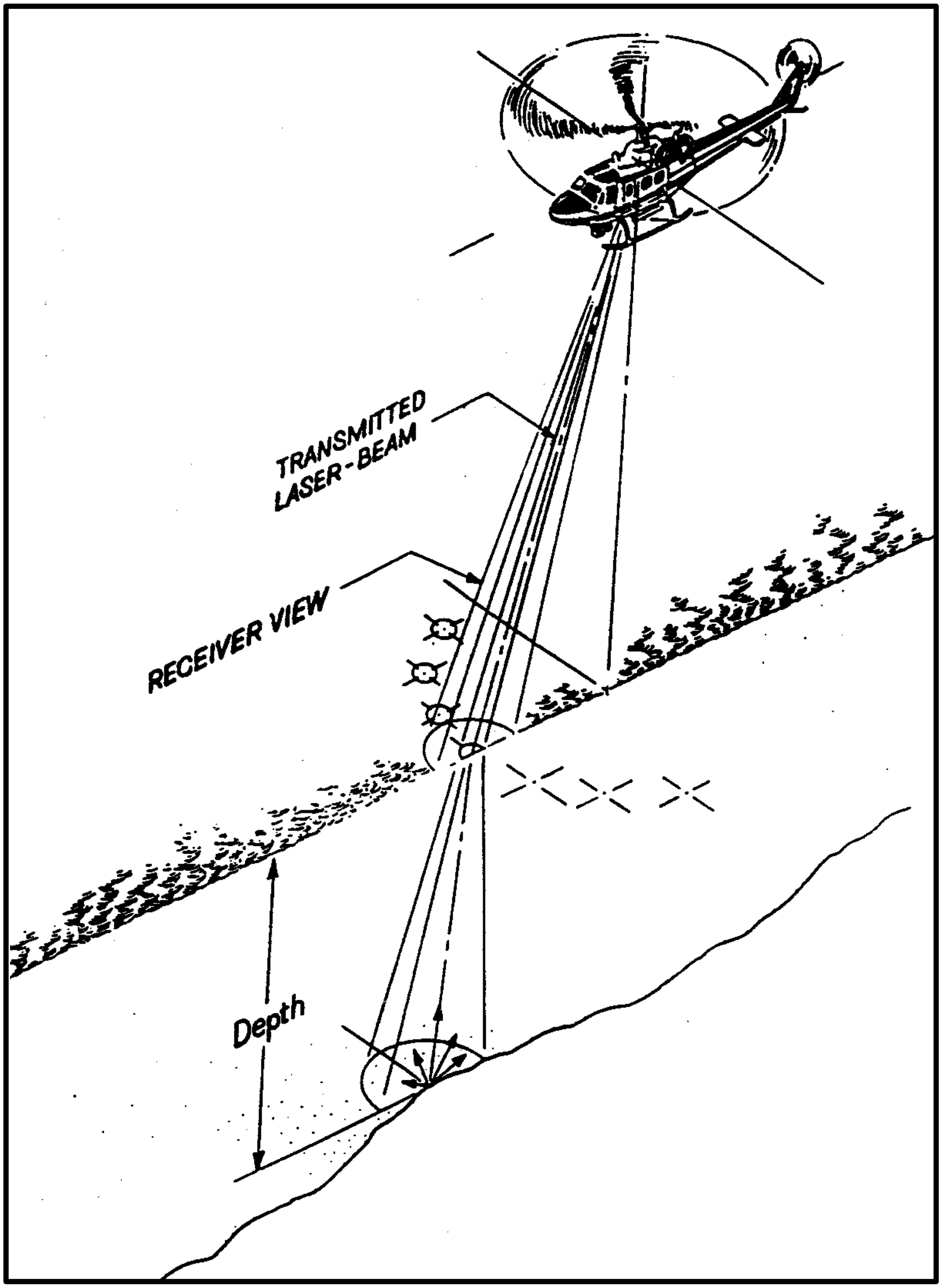

Figure 1: System operating principle (Courtesy Joint Airborne Lidar Bathymetry Technical Center of Expertise.)

in very clear waters to less than 10 meters in cloudy, turbid waters (Guenther, Eisler, Riley, and Perez 1996).

Airborne lidar technology offers a high rate of area coverage at high density to hydrographic surveying. However, there are many biases which must be taken into account and corrected in order to provide truly accurate depth measurements. This involves careful hardware design, post-flight data processing software, and the consideration of many environmental factors. Due to the large volume of data collected, it is beneficial for this analysis and error detection to be as automated as possible (Guenther, Thomas, and LaRocque 1996).

SHOALS System Technology

The Scanning Hydrographic Operational Airborne Lidar Survey system was designed and constructed by the Canadian company, Optech, Inc., for the Canadian Department of Industry, Science and Technology and for the Waterways Experiment Station of the U.S. Army Corps of Engineers. John E. Chance and Associates (JECA) of Lafayette, Louisiana currently operates the system, using either a Bell 212 helicopter, or a fixed-wing Twin Otter, flown by the National Oceanic and Atmospheric Administration (NOAA) Aircraft Operations Center (Guenther, Thomas, and LaRocque 1996). The SHOALS scanner transmits laser pulses in a 180 degree arc across the aircraft flight path (See Figure 2). Surveys are generally produced at a nominal 4 meter spacing; however, the survey speed and scan pattern can be adjusted to obtain either higher or lower sounding densities (Parson, Lillycrop, Klein, Ives, and Orlando 1997).

SHOALS is composed of an Airborne Data Collection System and a Mobile Data Processing Facility. The airborne system has three subsystems: the acquisition, control and display system (ACDS), the transceiver, and the positioning and auxiliary sensors. The ACDS provides real-time project depths to the survey operator (not corrected for tides, or waves, but

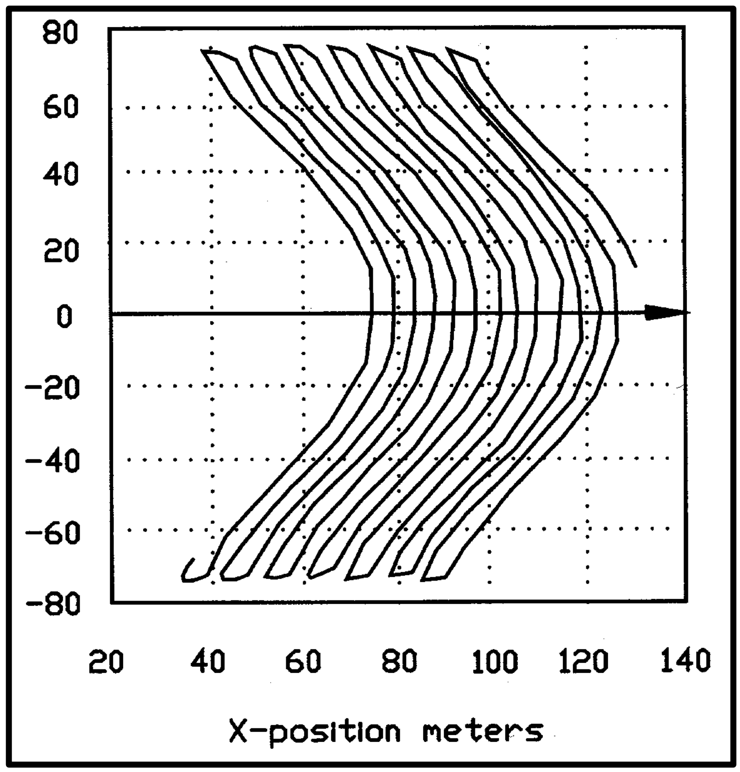

Figure 2: SHOALS scanner laser pattern (Courtesy Joint Airborne Lidar Bathymetry Technical Center of Expertise.)

accurate to within about plus or minus one meter). It is this component of SHOALS which also provides navigation information to the pilot (i.e. required altitude, speed, and position along a specific survey line). These updates allow for a survey of desired density to be completed (Lillycrop, Parson, and Irish 1996).

The SHOALS transceiver is mounted in a pod below the helicopter. Its main component is a Nd:YAG (neodymium:yttrium, aluminum, garnet) laser transmitter and receiver that operates at a 200 Hz pulse rate. The laser transmits at a wavelength of 1064 nm with an infrared (IR) output of 15 mJ, and a simultaneous frequency-doubled (532 nm) green output of 5 mJ. There are five receiver channels, three used for water surface detection, and the remaining two for detecting the sea bottom. The pod also houses the gyro-stabilized scanner that directs each pulse to the water's surface, and the inertial reference system which provides aircraft attitude information (Lillycrop, Parson, and Irish 1996).

The positioning and auxiliary sensors consist of a differential Global Positioning System (GPS) satellite receiver and a video camera. SHOALS currently achieves a vertical survey accuracy of plus or minus 15 cm (about 6 in), and a horizontal accuracy of two to three meters (6.5 to 10 ft) (Lillycrop, Parson, and Irish 1996). The video can be used to analyze anomalies which may be encountered during a survey. SHOALS can also collect topographic elevations in conjunction with water depths. In early 1997, work was begun to add an on-the-fly (OTF) kinematic GPS capability to the system. This carrier-phase tracking technology allows the altitude of the aircraft with respect to the ellipsoid to be determined (Guenther, LaRocque, and Lillycrop 1994). With the new vertical reference, the need for in-the-field tide measurements is reduced, and the extent of topographic measurements adjacent to the coastline may be expanded. The new technology also permits determination of accurate depths without the difficult wave correction algorithms that must otherwise be applied (Lillycrop, Irish, and Parson 1997 and Guenther, Thomas, and LaRocque 1996). Additionally, dual-baseline post-processed kinematic GPS allows for horizontal lidar accuracy of about 1 m (about 3 ft) (Watters and Wiggins 1999).

Post flight data processing is done in a mobile support facility housed in a forty-foot tractor-trailer (See Figure 3). The trailer travels site to site with the helicopter, conducting surveys. Waveform processing is done on a Sun SPARCstation System 10 with 160 MB of memory, 10 GB of disk space and an Exabyte 8500 tape drive. A Hewlett Packard HP735 with 148 MB memory and 2 GB of disk space is used for geographic modeling. SHOALS produces ASCII output files which can be used in geographic displays by commercial software packages (Guenther, Thomas, and LaRocque 1996).

Figure 3: SHOALS helicopter and mobile field unit (Courtesy Joint Airborne Lidar Bathymetry Technical Center of Expertise.)

As mentioned earlier, airborne lidar surveys must undergo numerous system and environmental bias corrections in order to provide accurate depth measurements. With the proper implementation of these adjustments, the survey depths can be brought within International Hydrographic Office (IHO) standards of accuracy. For example, optical and time delays must be determined, and the precise nadir angle of the beam must be known. The level of confidence in the output data is dependent on factors such as timing, angle calibration, the sensitivity of the signal detectors, the identification of false targets, and the removal of wave heights as necessary (Guenther, Thomas, and LaRocque 1996).

Two of the most difficult ALB problems to overcome are accurate detection of the air/water interface, and the amplitude dynamic range between surface and bottom returns. When wind speeds are sufficient to create capillary waves on the water’s surface, an IR channel can detect infrared signals reflected back to the aircraft. A separate IR channel acts as a “land/water discriminator.” In calm waters, SHOALS can use a channel in the red portion (645 nm) of the spectrum to detect the Raman volume backscatter return as an alternative surface locator. A flexible receiver, therefore, has channels in three wavelengths: green, IR, and red (Guenther, Thomas, and LaRocque 1996).

SHOALS actually has two green channels, two IR channels, and one red channel. This configuration is not only used to ensure accurate air/water interface detection, but also contributes to the dynamic range solution. The amplitude dynamic range is referring to the big difference between the strongly detected surface return and the very weak bottom return. These two signals reach the receiver within tens or hundreds of nanoseconds of one another and differ by six orders of magnitude in amplitude. It is difficult to digitize signals of this nature. Using two separate channels helps the situation. A high-gain, sensitive green channel is used for the weaker, deep returns and a "shallow," less sensitive green channel is used for stronger, shallow depths and for the surface return (Guenther, Thomas, and LaRocque 1996).

History of Airborne Lidar Bathymetry

Using an airborne, pulsed laser system for bathymetric measurements was an offshoot of an early 1960s effort to use lasers for submarine detection. The Naval Air Development Center (NADC) sponsored many theoretical and modeling laser studies in support of an anti-submarine program. Ongoing experiments were conducted at the Scripps Institution of Oceanography as well. The initial hydrographic applications were also militaristic in nature, searching for a means of remotely surveying hostile areas (Guenther 1985). In 1969, Hickman and Hogg of the Syracuse University Research Center wrote a paper confirming the ability of an ALB to detect nearshore bathymetry. By the early 1970s, airborne lidar surveying had been successfully tested by the U.S. Navy, by the National Air and Space Administration (NASA), by Canada, and by Australia (Guenther, Thomas, and LaRocque 1996).

The U.S. Naval Oceanographic Office field-tested the Pulsed Light Airborne Depth Sounder (PLADS) system in 1972. In the same year, NOAA began studies on the operational feasibility of airborne lidar hydrography. In 1975, a second-generation system was designed for NASA by Avco Everett Research Laboratory, Inc. It was called the Airborne Oceanographic Lidar (AOL). Eventually, NOAA and the newly formed Naval Ocean Research and Development Activity (NORDA) co-sponsored the project which was focused on the two areas of hydrography and fluorosensing. The NOAA/NORDA/NASA AOL experiment was established to determine the potential of airborne laser bathymetry (its accuracy and maximum penetration depth) and also to identify and quantify the effects of the many system and environmental variables and parameters involved. Much of the testing required specially designed Monte Carlo computer simulation software to support the fieldwork (Guenther 1985).

The first AOL test flights were conducted over Chesapeake Bay and the Atlantic Ocean near Wallops Island, Virginia in 1977. The test phase was concluded the following year. Although the AOL program was considered a great success and a breakthrough in lidar technology and understanding, it was terminated due to lack of funding in hopes that other groups, perhaps commercial, would take the lead in the development of ALB systems. During the same time frame, second generation systems were successfully tested in Canada, Australia, the Soviet Union, and Sweden (Guenther 1985).

NORDA under Defense Mapping Agency (DMA) sponsorship, decided to invest in a "dedicated helicopter-mounted bathymetric system" to be called the Hydrographic Airborne Laser Sounder (HALS). In 1979, the Avco Everett Research Laboratory, Inc. of Everett, Massachusetts (builder of the AOL) won the contract. Unfortunately, it was terminated before scheduled hardware delivery due to managerial and technical problems (Guenther 1985).

The early 1980s saw heightened worldwide interest in the science of airborne lidar. The Australians successfully flight tested their second generation scanning system and began design of an operational Laser Airborne Depth Sounder (LADS) system. The Canadian program which originated at the Canadian Center for Remote Sensing (CCRS) produced the first fully operational ALB system called the Larsen-500. It was supported by the Canadian Hydrographic Service. The Swedish developed a system called FLASH which is now operational and renamed Hawk Eye. The Soviet Union tested three, multipurpose research systems, and China was also conducting ALB research. It was during this same time frame that work began on the USACE's SHOALS program (Guenther 1985 and Guenther, Thomas, and LaRocque 1996).

ALB technology continued to mature throughout the eighties, and, by the 1990s, four systems were being used operationally: the USACE's SHOALS and the Swedish Hawk Eye (both helicopter borne in aircraft of opportunity), the Australian LADS (in a dedicated Fokker F-27 fixed-wing aircraft), and the Canadian Larsen-500 (flown in several fixed-wing aircraft). Use of airborne lidar is becoming increasingly popular, and other nations are expressing interest in the systems and their surveys. SHOALS has been flown in Mexico, and one of the two Hawk Eye systems has been operated in Indonesia (Guenther, Thomas, and LaRocque 1996).

SHOALS Operations and Capabilities

With SHOALS system development and operation, the USACE is working towards a long-term goal of encouraging the technology's growth in the private sector. It is hoped that other government agencies (besides USACE and NOAA) and the commercial survey industry will continue to expand and develop lidar survey capabilities (Lillycrop and Estep 1995). Unlike Canada's Larsen-500, SHOALS was not designed for large-scale chart production surveys covering hundreds of square kilometers. Rather, the system was developed to survey small projects, only a few square kilometers in size. In this capacity, it is estimated that airborne lidar bathymetry lowers USACE project survey costs by thirty percent. Additionally, there are fewer demands on personnel and equipment (Lillycrop, Parson, and Irish 1996).

Between March 1994 and June 1997, SHOALS surveyed over 125 projects, covering over 2250 square kilometers of U.S. coastal waters and lakes (Lillycrop, Irish, and Parson

1997). In its first two years of operation, SHOALS collected about 180 million soundings. A conventional single-beam acoustic system would require about 24 years to accomplish that task (Lillycrop, Parson, and Irish 1996). Originally, SHOALS was designed to meet USACE Class 2 survey requirements. However, the system currently meets both Class 1 survey requirements and IHO charting standards. High accuracy Class 1 surveys are required for dredge payment (Irish, Lillycrop, Parson, and Brooks 1994 and Irish, Lillycrop, and Parson 1996).

In early 1994, SHOALS was field-tested and its accuracy (horizontal/vertical) compared to a multibeam fathometer with favorable results. The following year, SHOALS surveyed 50 square kilometers of Tampa Bay. It took only 12 hours to collect the 5.5 million soundings with depths ranging from 10-20 meters. The same area was surveyed by the NOAA ship MT MITCHELL using its vertical-beam echo sounder. That mission required four months to collect 30,000 depth soundings. The MT MITCHELL survey points were mapped onto a digital terrain surface created from the higher density SHOALS sounding. The standard deviation of the depth differences was 0.2 meter, resulting from a bias which is accounted for in today's SHOALS measurements. The results of this test confirmed that SHOALS meets IHO nautical charting standards (Lillycrop, Irish, and Parson 1997 and Guenther, Thomas, and LaRocque 1996).

One of SHOALS' primary USACE missions is the monitoring of shoaling in navigation channels. The system can be used to quickly quantify dredging volumes. It is also useful in planning or evaluating beach nourishment projects. Due to the expense of beach quality sand, miscalculating project design volumes can be very costly. Traditionally, shore-normal profiles are surveyed in order to compute dredge or fill volumes. These profile lines are spaced every

30-300 meters along the coastline. Since the nearshore and beach may be highly irregular, it is easy to under or over estimate using this technique. In fact, volume error has been seen to increase nearly linearly as profile spacing increases. At many test sites, the economic benefits of higher density SHOALS bathymetry has been proven. However, the density requirements for a survey are dependent on many different factors, to include the degree of along-shore variability and the intended application of the survey (Irish, Lillycrop, and Parson 1996).

Another important SHOALS mission is the mapping, modeling and charting of coastal waters. In the Florida Bay Restoration Program, the system was used to define water circulation patterns in Florida Bay, a large, shallow-water estuary. This area is difficult to survey due to shallow depths and risk to the environment. A conventional approach would be slow and resource demanding. However, SHOALS was able to quickly produce high-resolution surveys which met the specifications of the environmental conditions (Parson, Lillycrop, Klein, Ives and Orlando 1997).

One important advantage of the remote airborne collection of bathymetry is the ability to collect data in very shallow waters without danger of running aground. SHOALS has conducted nautical charting missions in many different areas including Miami Harbor and Port Everglades, Florida; Elwood and Gaviota, California; and the Yucatan Peninsula, Mexico. The survey off the coast of Mexico was conducted in early 1996. It involved 56 days of data collection over an area of 800 square kilometers and was successful in locating a previously uncharted shipwreck (Lillycrop, Irish, and Parson 1997).

In order to enhance ALB operations in the United States, a more thorough knowledge of nation-wide water clarity conditions is required. Water clarity is the main operational constraint for SHOALS or any ALB. Lidar surveys require much more detailed mission planning than the traditional sonar survey to ensure success and to remain cost-effective (Lillycrop, Parson, and Irish 1996). As discussed earlier, the maximum penetration depth of laser pulses traveling through the water column is determined by the physical effects of scattering, absorption and refraction. Maximum penetration is directly correlated to the maximum depth of bottom detection. Various environmental factors in the survey area will affect these quantities: incident sun angle and intensity, water turbidity, radiance of the bottom material, type and quality of suspended organic particles and sediments (Parson, Lillycrop, Klein, Ives, and Orlando 1997).

In general, SHOALS can sense the seafloor to depths of two to three times the Secchi depth, depending on the composition of suspended material. An oceanographer's tool called a Secchi disk is used to measure Secchi depth. This white disk is suspended from a line and lowered through the water column until it disappears from sight. It is that point which determines the Secchi depth (Lillycrop and Estep 1995). The USACE has realized the value of a seasonal Secchi depth database as a means of monitoring water optics in prospective SHOALS deployment regions. In the early 1990s, each district office routinely began recording these optical measurements. Now, these data serve as a mission planner tool which can be used to schedule optimum times for surveys to be conducted in each area of interest (Estep, Lillycrop, and Parson 1994).

Even without planning, airborne lidar can be an extremely useful tool in assessing coastal conditions following major storm events. It is rapidly deployable and capable of collecting both bathymetric and topographic data needed for damage assessment. In mid-October 1995, Hurricane Opal hit the western Florida panhandle resulting in significant coastal changes. SHOALS was used to quickly detect and chart the dangerous shoaling and severe erosion caused by the storm. The system was also used to determine the level of coral reef damage after a vessel grounded at Maryland Shoal in the Florida Keys (Irish, Thomas, Parson, and Lillycrop 1996 and Lillycrop, Irish, and Parson 1997).

To enhance operational capability, SHOALS may be operated simultaneously with other sensors such as an imaging spectrometer, or hyperspectral scanner (HSS). An imaging spectrometer provides many spectral channels of narrow bandwidth to collect unique reflectance spectrums used to identify the materials from which light is reflected. Together, SHOALS and an HSS can classify and quantify the environment below the water's surface. In 1985, the Environmental Research Institute of Michigan (ERIM), the U.S. Navy, and Borstad & Associates with TerraSurveys, Ltd. flew an active/passive scanner called M8 near Cat Cay in the Bahamas. Lidar was successfully used to calibrate the multispectral imagery. Six years later, Borstad & Associates and TerraSurveys, Ltd. used CASI (the Compact Airborne Spectrographic Imager) to collect hyperspectral imagery of Lake Huron. These data were used in conjunction with lidar soundings surveyed by the Larsen-500 over the same area. The combined depths and imagery provided a high-resolution digital model, detailed enough to show sand waves which are not detectable by lidar alone (Lillycrop and Estep 1995).

The most important advantage of using HSS imagery with SHOALS data is the ability to extract information about seafloor type. If *a priori* spectral signatures are collected for different bottom types such as sand, mud, and seagrass, the HSS and SHOALS fused data can be used to identify the seabed at various depths. A survey mission of this sort could evaluate environmental impacts before an engineering project is begun. The loss or gain of resources might also be monitored with this type of survey (Lillycrop and Estep 1995).

From nautical charting to sediment transport studies, and from beach monitoring to storm damage reconnaissance, SHOALS has proven a versatile and effective survey instrument. Its high-density data can be useful in detecting features that might easily be overlooked by more traditional survey techniques. With proper planning, the system provides a timely, cost-effective, quality product to its user community. As technology advances, SHOALS missions will no doubt diversify, and its capabilities improve.

REDUCING THE DENSITY OF BATHYMETRIC DATA

The Concept of Data Density Reduction

Reducing the density of data is a useful step when the volume of a data set is so large and/or complex that it prevents, or inhibits, analysis. It is often necessary when attempting to portray data at a smaller scale. Condensing the data set down to a more manageable volume can be done by eliminating those points with little new information content. The data can then be expressed as a subset of the original set. If done properly, data reduction will be accurate and effective. Accurate in that the new set captures the meaning of the original set, and effective in that it is an easier version with which to work (Brady and Ford 1990).

Much of the literature on data density reduction or "thinning" algorithms concerns their application to image processing. Generally, the method involves the movement of a rectangular array over each point in an image in order to identify those pixels which represent the outline of the characters or objects in the image. The technique is called skeletonizing. It is an effort to reduce data by keeping only the significant points and preventing any real information loss. The motivation of image thinning (especially in character recognition) is to develop a line drawing of an originally "thick" image. The fewer points to be analyzed in the recognition phase, the better (Deutsch 1972).

A point whose removal does not change the topology of the image of which it is a part is referred to as a “simple point”. Points that **do** impact topology are sometimes called “mass points”. The concept of simple points is, therefore, essential to image transformations, such as density reduction algorithms, which strive to preserve topological features. Bertrand suggests that, in thinning an image, one should delete simple points which are not end points. The simple point condition prevents an undesired change in the topology of the image, while the non-end point criterion saves useful information pertaining to the shape of the object (Bertrand 1994 and Bertrand 1995).

In applying approximation to surface modeling, the problem is how to sample only significant data points from a set of surface points. Ideally, highly curved areas should be sampled densely, while less curved areas are sampled more sparsely. This process is important to object recognition in computer vision and to surface modeling at various scales (Li 1993).

In his paper "Optimal Approximation in Automated Cartography", Wigand Weber of the Institut für Angewandte Geodäsie, Frankfurt describes the method of cartographic generalization as a model of mathematical optimization. He concentrates on the process of transforming the contents of a map into a version appropriate for a map of smaller scale. This involves removing some of the original map's information content, but retaining the information required for the new scale. Actually, the same concept is employed when recording natural topography on a map of any scale that is not one-to-one (Weber 1977).

Generalization is considered to have been performed well when the natural topography of an area may be optimally recovered from the limited information portrayed on the map. Cartographic generalization may involve the selection of objects to be omitted, the simplification of lines or features, the combination or summarization of objects, and/or the displacement of objects. Of course, some of these options may not be suitable for specific data types. In continuous data of equal type the "information trend" is conveyed to the map by the low frequencies in the data, while the majority of the information content is contained in the higher frequencies. Various methods can be used to detect trend and content, to include Fourier Transformation, least-squares interpolation, and gliding weighted arithmetic mean technique, also called convolution or low-pass filtering (Weber 1977).

Hydrographic Data Density Reduction and SHOALS Data

In reducing the density of airborne lidar bathymetric data, the goal is the efficient reduction of a very large data set to a “hydrographically representative,” but much smaller, subset. Often features of hydrographic interest are spatially oversampled, and the same terrain could be effectively modeled with much less data. Selecting a subset of survey data is often a necessary step prior to contouring in the charting process. In addition, data processing packages can better handle these decreased volumes of data (Guenther, Eisler, Riley, and Perez 1996).

The National Imagery and Mapping Agency (NIMA) uses several different decimation, or thinning, algorithms to reduce the density of the bathymetric data required for various products. In general, these methods rely on a process called "binning", that is, subdividing the data set of x, y, and z points into equally dimensioned map cells. Binning can also be envisioned as draping a "virtual grid" of cells (or bins) over the data, so that the points appear clustered within the various cells. Bin dimensions are generally chosen to represent the highest resolution the data support. (This resolution is related to the Nyquist frequency which can be determined through spectral analysis of the data.) A different algorithm uses a bin size which enables the number of output values (one per bin) to equal the number of soundings required in the end product. The value of each cell can also be determined by a variety of methods. Sometimes an average of the z-values of all points within the cell serves as the output z-value. This is a simple gridding technique, with the x and y values being those of the cell's center. Another algorithm might choose the x, y, and z values of a representative point within the cell as the output x, y, and z values. Often, in hydrographic applications, the least deep of the soundings in each cell is chosen for output, a practice termed shoal-biasing.[1]

Other organizations may use different methods to reduce the amounts of bathymetric data with which they work. There are off-the-shelf (OTS) gridding software packages which enable a surface model to be represented at any given resolution. However, they generally use averaging and interpolation to select the output values for each grid cell, and, therefore, lose the exact values of the original raw data set. Depending on the accuracy requirements of the user, these products could be quite suitable.

Although there are no specifics published on the thinning method employed by the LADS MkII system, the Australians have reportedly been using a proprietary thinning algorithm for over ten years. According to the technical director for the system, the method considers the variables of depth, confidence of depth, spatial density, relative position (to other soundings and

---

[1] Lambrecht, M., 1998. "Decimation White Paper," unpublished e-mail forwarded to the author by NIMA on 13 March 1998.

edges, etc.) and the hydrographic operator nominated scale of the survey. LADS does not use a gridding method to thin data.[2]

In 1996, an algorithm for thinning SHOALS data was published by Guenther, Eisler, Riley, and Perez of NOAA. In the paper, the authors describe a method of "Data Decimation for Airborne Laser Hydrography." It avoids the binning technique and, instead, moves linearly along each collected laser swath of data, one swath at a time. The algorithm detects areas, or neighborhoods, within the data swath which are sufficiently flat, and may be represented by fewer points. The output file is composed of a representative subset of data points selected, and is reduced from its original size by, roughly, a factor of ten (Guenther, Eisler, Riley, and Perez 1996). It is essential to note, however, that an accuracy assessment of the reduced data set relative to the original survey is not included in the paper.

In the Guenther method, the user selects the horizontal and vertical parameters that determine the extent of the local neighborhoods within each swath. There are other input options which allow for smoothing and shoal-biasing, as well as the inclusion of strategic local "peaks" and "deeps." The decimation routine moves linearly through the input data, retaining only points that represent a significant change in vertical depth or horizontal distance with respect to the given user input thresholds. There is a minimum of free memory required to run the program, since only one data block of 200 points (representing a single swath) is processed at a time (Guenther, Eisler, Riley, and Perez 1996).

---

[2] Perry, G., 1998. "LIDAR Thinning Algorithm," unpublished e-mail correspondence from Technical Director of LADS Corporation Ltd. on 19 June 1998.

The major advantage of the Guenther thinning algorithm over gridding techniques is that the process tends to be more efficient, although a quantitative accuracy assessment is not included in the article. A drawback to the method is that the points of an adjacent swath are not considered during the analysis of a particular line of data. The thinning is done in only a one-dimensional direction, rather than by considering all surrounding nearby points, as with a two-dimensional (2-D) analysis technique. It can be argued that the retention of points on either side of a single swath (which might otherwise be removed by a 2-D technique) provides useful cross-checking data to the final output set. However, the reduction ratio is limited by the linear data stream processing approach (Guenther, Eisler, Riley, and Perez 1996).

A variation of the Guenther method would be to incorporate the concept of breaklines, or breakcurves, into the linear analysis of each swath. If the knowledge of the terrain between retained data points in the thinned set was maintained, by the use of appropriate equations, a Triangulated Irregular Network (TIN) might be used to effectively model the network of data points and lines. In other words, if all but two points were removed from a swath line of constant slope, those two points would adequately map the terrain below the swath, as long as the equation for the line between them was included in the model. The same technique could be applied to three or more points using the equation for the curve they represent.

MATERIALS AND METHODS

A Data Density Reduction Algorithm

The data density reduction algorithm presented in this section was written in the Avenue programming language. It is designed for use on an ArcView GIS equipped with the Spatial Analyst extension. The actual code is included in Appendix A. The thinning process avoids gridding the input data points, and results in an output which is a subset of the original raw data set.

The first step in developing the algorithm was to consider the method in which the data would be accessed. As explained previously, SHOALS data consists of a collection of x, y, and z coordinate sets. The x and y represent a horizontal position on earth, and the z represents a vertical elevation with the vertical datum (zero elevation) generally being mean lower low water (MLLW). (Therefore, negative z-values in water correspond to are depths, or distance below MLLW.) Once plotted on an ArcView screen, the order in which the depths were collected by the scanning laser is not obvious. One merely sees a number of irregularly spaced points on a white background. However, this "shapefile" of visual points, geographically positioned in the view, is attached to a table that consists of all of the x, y, and z coordinates for those points. The table is linked such that, when a point's coordinates are selected (meaning a record is highlighted), the corresponding point is highlighted on the view. In Figure 4, note that the yellow (highlighted) records in the table of point values correspond to the yellow points in the view. By systematically running through each record in the associated table, every point in the view is considered.

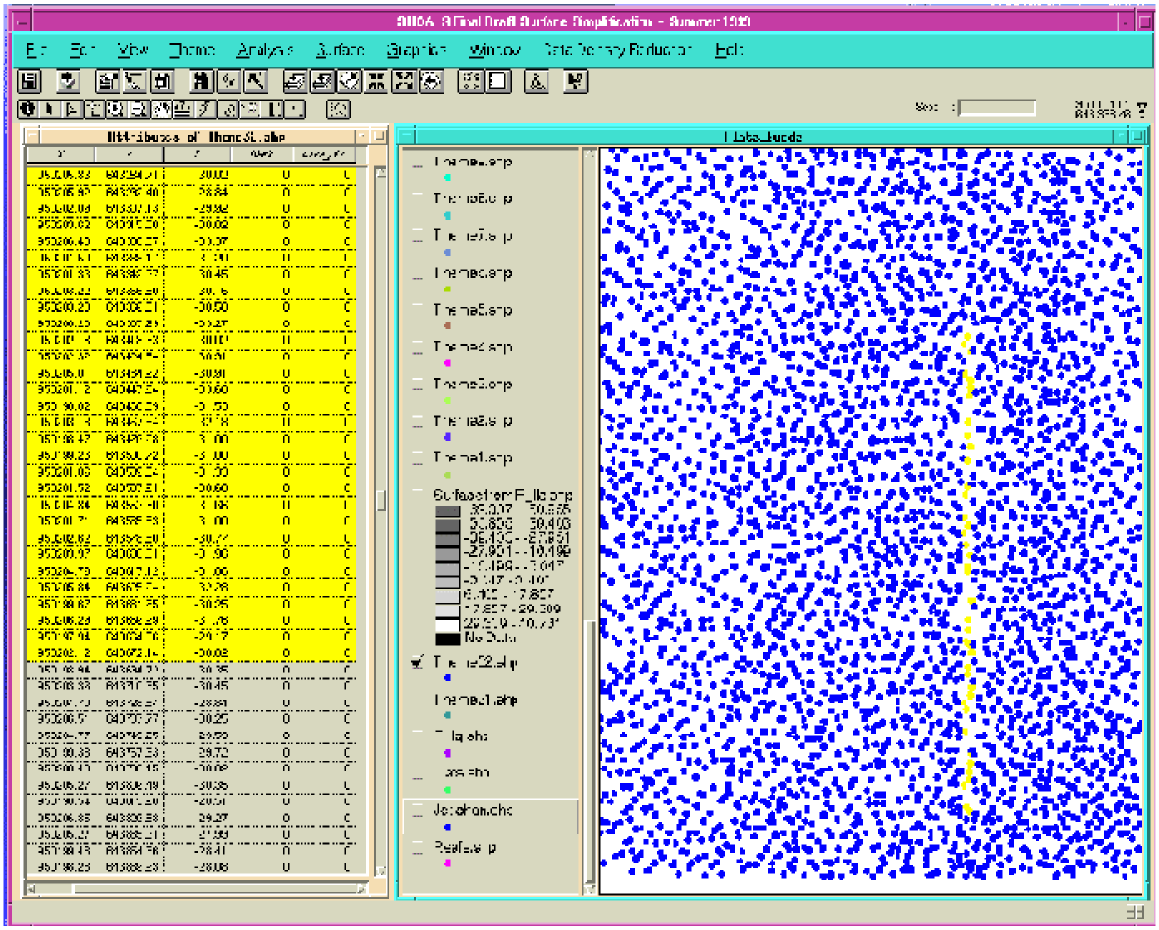

Figure 4: Highlighted Records in Table Correspond with Yellow Points in View.

The effectiveness of the DDR algorithm is highly dependent on two parameters which must be input by the user: radius and tolerance. The radius is a horizontal distance value (in the measurement units of the data) which determines the neighborhood around a given point. It should be selected based on the variability of the terrain being considered. The points within a defined neighborhood are then compared to the center point in the thinning process. The tolerance is a vertical distance value (in the measurement units of the data) which defines redundancy. If the z-value of a point within the neighborhood differs from the z-value of the center point by less than the stated tolerance, then it is considered redundant, and one of the two points will be removed. In order to maintain accuracy throughout the thinning process, tolerance values should be chosen so as not to exceed the vertical measurement error of the data. Using statistics, it can be determined whether or not the parameters selected were sufficient to achieve the density reduction sought, while maintaining the accuracy required.

ArcView makes it easy to access the neighboring points within a specified distance of a selected point. The process is called theme-on-theme selection, and it is this software capability which enables the thinning algorithm to perform as follows:

1. Program accesses a record (operating point, or OP) in a table of all data points.
2. If the record is "marked," it moves on to the next record. (Go to step 1) ("Marked" means that the point has been designated for removal.)
3. The program finds all records for the points within a specified radius from the OP. (These points in the "neighborhood" are called test points, or TPs.)
4. Each TP record is accessed, and, if the record is "marked," the program moves on to the next TP.
5. If the TP record is unmarked, its Z-value is compared to OP's Z-value.
6. If the absolute value of the difference between Zop and Ztp is greater than or equal to a given tolerance, then the program moves on to the next TP. (Go to step 4)

7. If the absolute value of the difference between Zop and Ztp is less than the given tolerance, then the two Z-values are compared.
8. If Zop < Ztp, then Zop is "marked" and, if Ztp <= Zop, then Ztp is "marked." (By marking the deeper soundings for removal, the data becomes shoal-biased.)
9. Once all of the test points in the neighborhood have been analyzed, the program moves to the next OP in the table of data. (Return to step 1)
10. After all OPs have been processed, the unmarked records are printed to a new file. This file of points is the reduced data set.

A flowchart which diagrams the process described above is presented in Figure 5. It is a relatively simple approach to identifying redundant information. Of course, the definition of redundant is determined by the user's selection of radius and tolerance.

A Comparison Technique

Once the thinning mechanism was established, a statistical method for evaluating its effectiveness was developed. By fitting a surface to the raw data set, and then fitting a different surface to the less-dense data set, the difference between the two can be identified. The use of grids in surface generation was determined to be acceptable for this “comparison phase”. In other words, grids are not being used to modify the actual data sets, but rather to assess their accuracies.

The inverse distance weighted (IDW) interpolation technique was employed to generate a surface, based on the actual points in a given data set. With this method, a z-value is

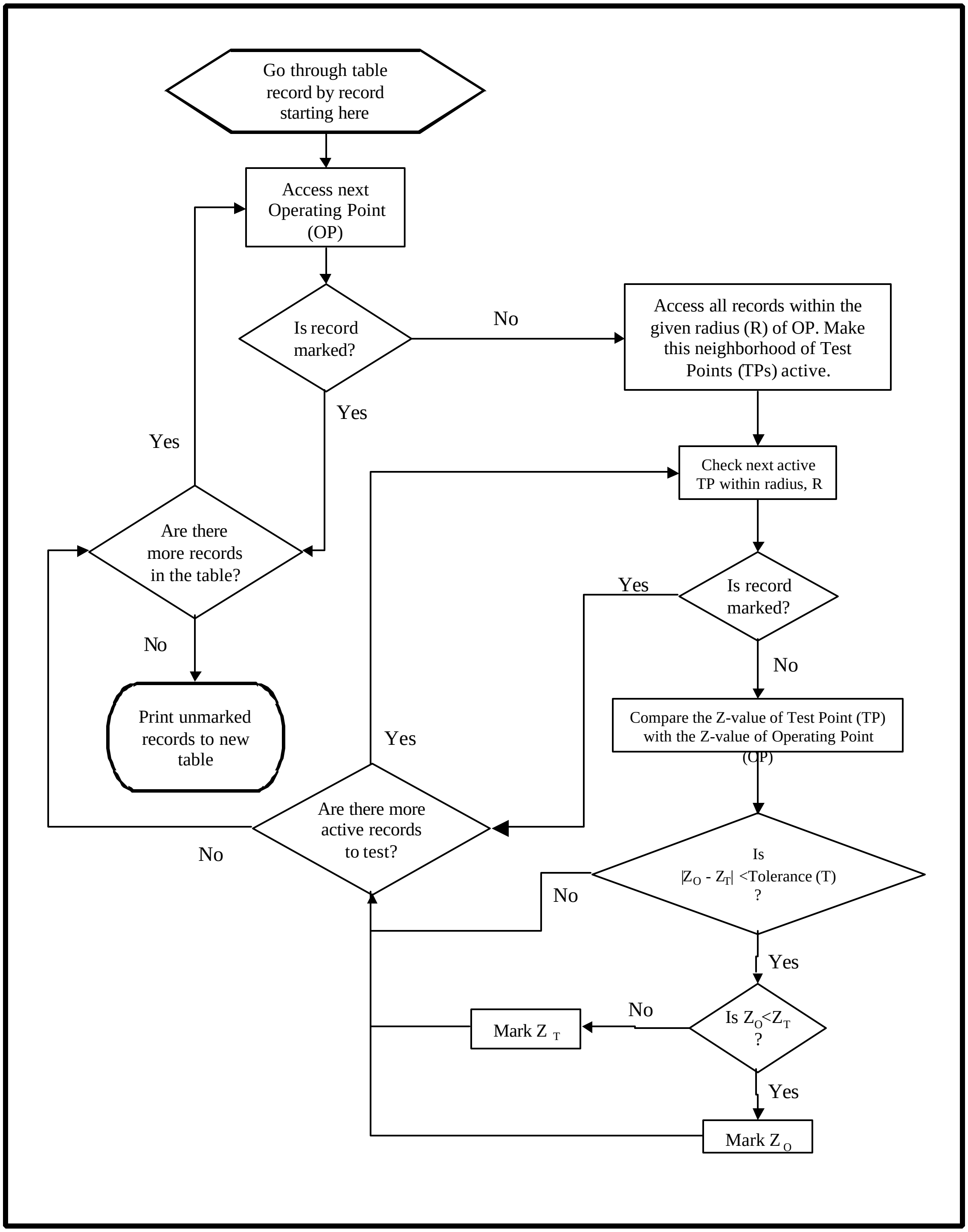

Figure 5: Flowchart of the Data Density Reduction Algorithm.

interpolated for each cell across the surface, using a weighted-average of the z-values of twelve closest neighboring data points. The “dense” surface, created from the original complete set of raw data points, is accepted as **truth**. Then alternate surfaces created from more “sparse” subsets of the raw data can be compared to this “truth” surface. It is desirable that the residuals, or the differences between the two surfaces, be as small as possible. The program computes the root mean square (RMS), or the sum of the squares of the differences along the two surfaces. A standard deviation which removes the mean from the RMS is also computed in an effort to isolate any obvious biases in the data density reduction process.

The Data Used to Test the Algorithm

Testing began with three sets of representative SHOALS data taken from along Florida’s eastern coastline: Flats, Jet-Chan, and Reefs. The data are on the horizontal datum of North American Datum (NAD) 1983 Florida State Plane East. The z-values reference the mean water surface and are corrected to “mean lower low water” (MLLW) through the use of a tide corrector (Guenther, LaRocque, and Lillycrop 1994). The Flats data set is a sample of flat underwater terrain consisting of 20,565 points having z-values that range from -63.39 ft to 60.93 ft. The Jet_Chan data represent a dredged channel between jetties and consists of 29,530 points with a z-value range from a -172.21 ft minimum to a 49.93 ft maximum. The Reefs data set is comprised of 29,170 points with z-values ranging from -112.27 ft to -61.45 ft (See Figures 6, 7, and 8).

Each of these three main data sets were then divided into slightly overlapping quadrants in order to decrease the computation times and to try to isolate patterns. For each area these

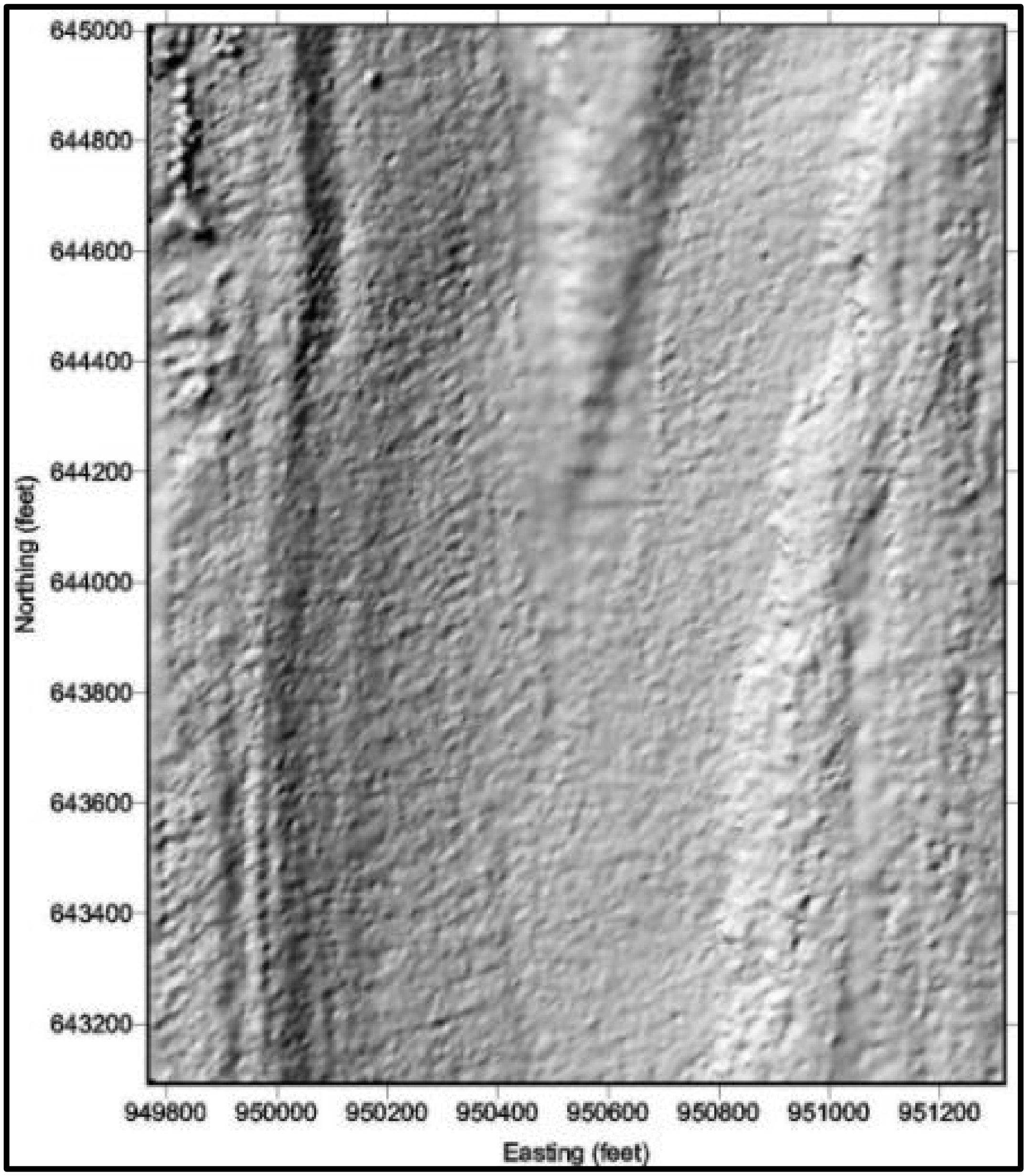

Figure 6. The Flats data set in shaded relief (Florida State Plane East NAD 83).

"quads" are called, the Upper Left (UL), the Upper Right (UR), the Lower Left (LL) and the Lower Right (LR). Because the three sample areas were surveyed at different densities, the computation times and the effects of the parameter values on these subsets is varied. However, definite patterns can be seen and conclusions may be drawn.

The effectiveness of the DDR algorithm was compared to that of two simplistic data density reduction methods. The first was called Systematic and, as its name suggests, deleted every n[th] point to allow for the "systematic" removal of a given percentage of the data. The second was called Random and, again allowed for the removal of a precise amount of data, but, in an arbitrary, rather than a systematic, manner. Once the DDR algorithm was employed with a parameter (radius and tolerance) pair, the data set was reduced to a given number of points.

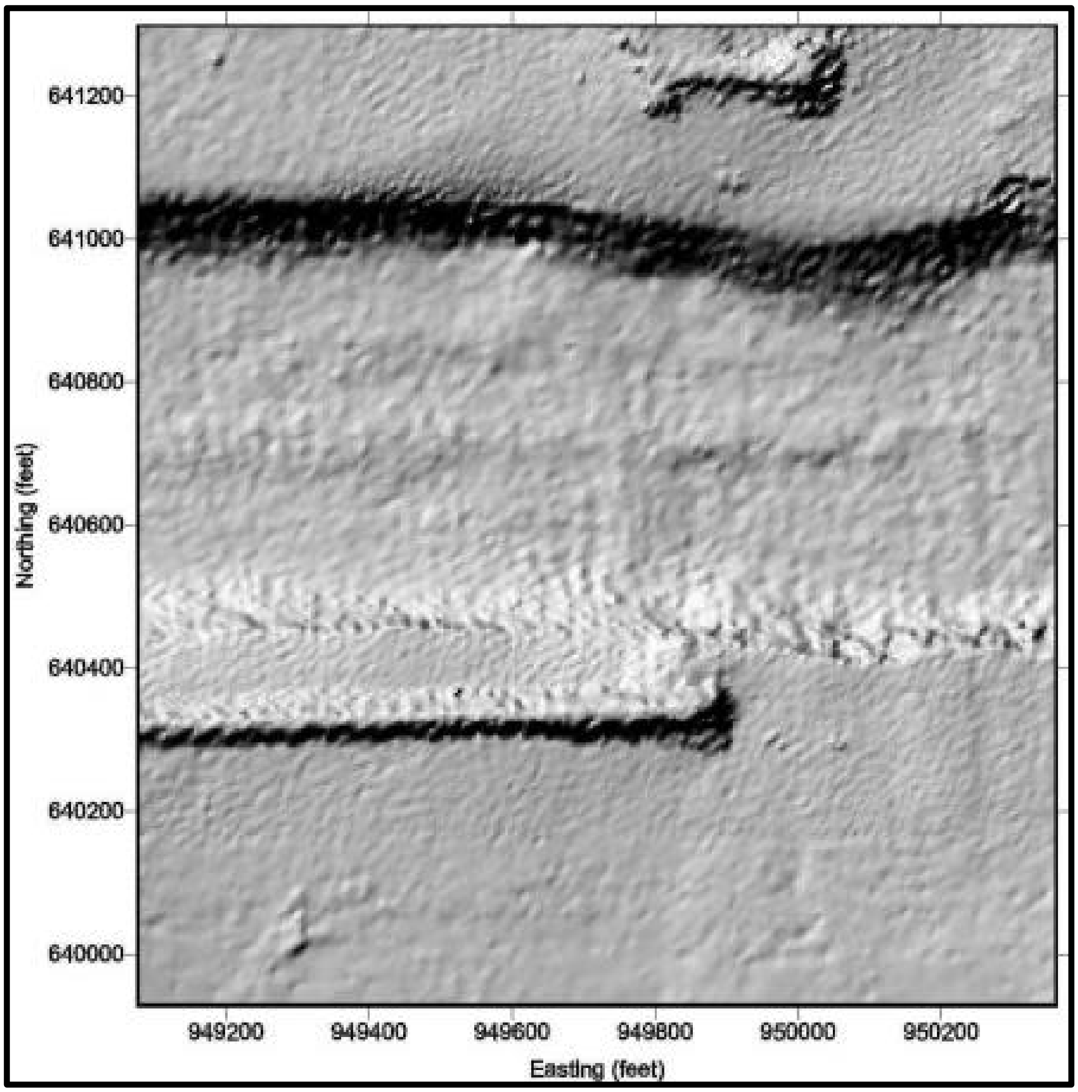

Figure 7. The Jet-Chan data set in shaded relief (Florida State Plane East NAD 83).

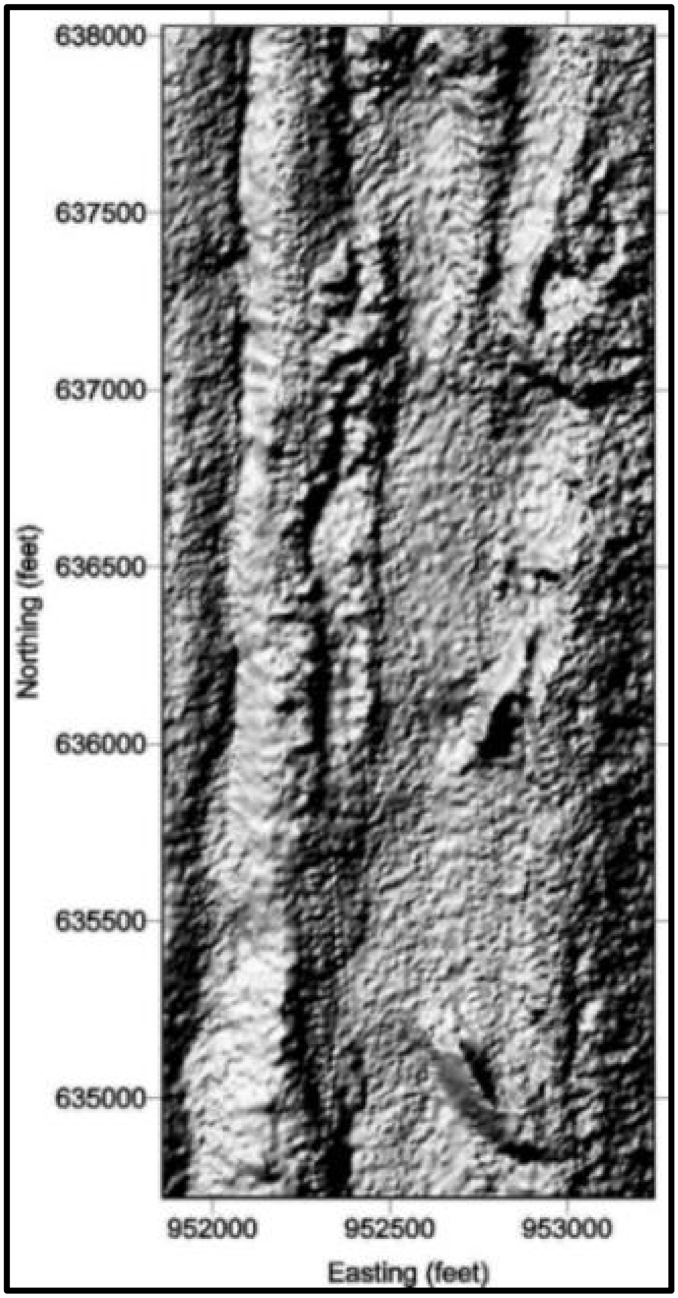

Figure 8: The Reefs data set in shaded relief (Florida State Plane East NAD 83).

Using both the Systematic and Random methods, the original data set could then be reduced to the same number of points for ease of comparison. Finally, since neither the Systematic, nor the Random, method of data density reduction is particularly practical, a further statistical test was conducted in an effort to demonstrate the resulting spatial accuracy of the algorithm. This was a chi-square test in which the RMS produced by the DDRA thinning process was compared to the vertical measurement error in the data.

Within each of the 12 quadrants of SHOALS data, nine experimental data density reduction runs were conducted. Tolerance values were kept within the stated vertical measurement error of the SHOALS system (+/- 0.5 ft). The radius values were chosen primarily in an effort to keep RMS values low (below 0.5 ft when possible), while reducing data density to the maximum extent. The radius/tolerance (in feet) combinations for these runs were as follows:

r = 30, t = 0.3	r = 20, t = 0.3	r = 10, t = 0.3
r = 30, t = 0.2	r = 20, t = 0.2	r = 10, t = 0.2
r = 30, t = 0.1	r = 20, t = 0.1	r = 10, t = 0.1

In each test quadrant, the above parameter pairs were first used to thin the data by means of the programmed DDR algorithm (DDRA). Next, the Systematic thinning method was used to thin the data to an equally reduced level. With the "Run Statistics" option, the original raw data was compared to the DDRA thinned data and to the systematically thinned data respectively. (The RMS value for the DDRA is referred to as RMS_D, while the RMS for the systematically thinned data is called RMS_S.) Random thinning was applied to each data set using only two of the parameter pairs (r=30, t=0.3 and r=10, t=0.1). This is because preliminary test runs indicated that the Random method would consistently yield a higher RMS than the algorithm. Again, the data sets were reduced to the same level as that produced by the

DDRA. Following the same routine as in the systematic comparison, RMS_D was compared with the RMS value for the randomly thinned data (RMS_R).

The Fisher Hypothesis Test

The RMS values generated through the techniques outlined in the previous section were then used to calculate the test statistic for a Fisher hypothesis test:

Null Hypothesis (H_O):	Systematic (or Random) method is as effective as, or more effective than, the DDRA
Alternate Hypothesis (H_A):	Systematic (or Random) method is not as effective as the DDRA

Using symbology, this concept can be expressed as follows:

H_O: $RMS_S <= RMS_D$ or $RMS_S / RMS_D <= 1$

H_A: $RMS_S > RMS_D$ or $RMS_S / RMS_D > 1$

(The same logic would apply if RMS_S were replaced with RMS_R.)

Since the RMS is being calculated across the thousands of points which comprise the surfaces being fit to the data, the "N" values for the hypothesis tests are extremely large (on the order of 10^5 to 10^6). In each case, the value of N, which represents the number of points in the population, is equal to the number of rows times the number of columns in the surface being fit to the data set. The degrees of freedom (DF) used to determine the F-statistic in a Fisher distribution table is a value equal to "N-1" for both the numerator and denominator. Most tables publish F-statistic values based on DF ranging from one to 120. The tables then jump to infinity and declare an F-statistic of one where the degrees of freedom for both the numerator and denominator equal infinity.

For each of the twelve sample sets of data, the DF values for numerator and denominator are so large that the F-statistic may be rounded to one. This was confirmed by using the University of California Los Angeles (UCLA) statistics web site[1] and entering the exact DF value to get the actual calculated F-statistic for a probability of 0.95 and a significance level of 0.05. The ratio of the RMS values (RMS_S / RMS_D or RMS_R/RMS_D) is called the test statistic. If this calculated test statistic is greater than the F-statistic of one, then the null hypothesis may be rejected. In that case, there is a 95% certainty that the alternative hypothesis is true. If the null hypothesis is not rejected, we conclude that the data do not provide sufficient evidence to support the alternative hypothesis.

The Chi-Square Hypothesis Test

The overall merit of the data density reduction algorithm was examined by use of the chi-square (χ^2) test. This test was deemed appropriate since the population was considered to be normally distributed. Normality was determined due to the large number of samples, N, being used, and because no significant bias was observed when RMS was compared to standard deviation.

---

[1] The UCLA statistics site is called "F Distribution CDF Calculator" and is located at the following universal resource locator (URL):
http://www.stat.ucla.edu/calculators/cdf/f/fcalc.phtml

The hypothesis test was based on the value of 0.5 ft, i.e. the stated vertical measurement error of the SHOALS system. If a combination of input parameters resulted in a thinned data set with an RMS not exceeding the 0.5 measurement error, then the test run was considered successful. The test was set up as follows:

Null Hypothesis (H_O): DDRA exceeds system measurement error

Alternate Hypothesis (H_A): DDRA performs within system measurement error

Using symbology, this concept can be expressed as:

H_O: $RMS_D >= 0.5$

H_A: $RMS_D < 0.5$

By using a left-tailed design, rejecting the null-hypothesis holds more weight. If H_O is rejected using a significance value of 0.05, then there is a 95% certainty that the alternative hypothesis is true. That means it is very likely that the DDRA does not bring the data outside the expected measurement error.

For each of the twelve quadrants of data, the χ^2 test statistic was calculated using all of the nine input parameter combinations. The χ^2 test statistic in each case was:

$$\chi^2 = DF\,(RMS_D)^2 / (0.05)^2$$

The χ^2 critical value is generally found in a chi-square distribution table by inputting the DF and a significance value of choice. However, many published χ^2 tables only go up to a DF of thirty. As with the Fisher test, the UCLA statistics web site was able to provide critical values for the extremely large degrees of freedom being considered. For each data run, the determination of whether or not to reject the null hypothesis is indicated in the final column of the three data summary tables in the Results and Discussion chapter. The value in this column is a "threshold RMS" determined by solving for the RMS in the test statistic, based on the critical value:

$$RMS_T = SQRT\ [(\text{critical value})(0.05)^2 / DF]$$

If RMS_D is less than RMS_T, then the null hypothesis is rejected and there is a 95% certainty that the DDRA method falls within measurement error.

Application Design

The data density reduction method was programmed with Avenue scripts imbedded in an ArcView project. It is a spatially visual tool and an ArcView "view" window must be active in order to see and to use the specially designed interface. There is a pull-down menu labeled "Data Density Reduction" with various options that lead the user through the thinning process (See Figure 9). The menu also includes a short explanation of the program under the title "About Application."

The most basic way of using the program is to follow the first three steps as labeled in the "Data Density Reduction" menu. In step one, the user is prompted to select an area of interest (AOI) from the data set. This can be either the entire set of points from the shapefile, or a specific area over which the mouse is dragged. Step two is the thinning process, "Run DDR" in the pull-down menu, which requires the user to input a radius value and a tolerance value in feet. After each record is analyzed and, either passed over, or, "marked" for removal, the points corresponding to the unmarked records are added to the view as a new theme. With the third menu step, "Run Statistics", the user is able to statistically compare two active point set themes. The program generates surfaces to fit each theme, finds the differences between the

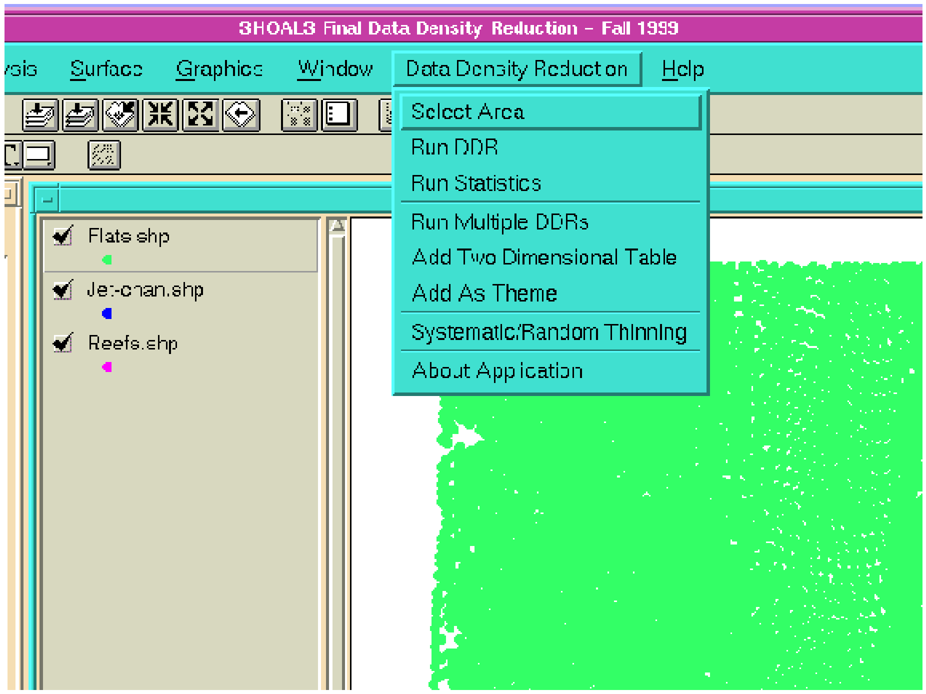

Figure 9: Data Density Reduction pull-down menu

two surfaces, and then creates a table with the RMS, the standard deviation and the mean values computed.

Other options on the pull-down menu allow for input of a combination of radius and tolerance values with which to experiment on a given set of point data. By using the program in this manner, the user can select that combination which produces an acceptable RMS and an acceptable percentage of data thinning. With the "Run Multiple DDRs" menu, one may input as many tolerance/radius pairs as desired for a given point theme. A table of computations from the multiple runs will be generated. The "Add Two Dimensional Table" menu choice allows the user to input numerous radius values and tolerance values separately (rather than in pairs). From these inputs, two two-dimensional tables of information are derived--one table for RMS, and one for DDR percentages (See Figures 10a and 10b). These tables are formatted with the input radii as column headings and the input tolerance values as row headings. Each entry in a table, therefore, corresponds to a particular row (tolerance) and column (radius). The RMS

(a)

RMS1.dbf

Tol	R10	R20	R30
0.1000	0.0853	0.2275	0.3178
0.2000	0.1276	0.3429	0.4889
0.3000	0.1703	0.4209	0.5961

(b)

pcent1.dbf

Tol	R10	R20	R30
0.1000	97.4144	88.6347	81.7708
0.2000	94.7545	77.7902	65.9040
0.3000	92.5037	70.2009	55.8408

Figure 10: Two-dimensional tables produced with ArcView application. A) RMS Values; b) DDR Percentages

values are the result of thinning by each radius/tolerance combination as compared to the original complete data set. The values in the percent-thinned table are simply the number of points in the thinned data set divided by the number of original points, times one hundred to get a percentage.

The Use of Fourier Analysis to Enhance the Technique

Earlier it was noted that the operators of the SHOALS system are seeking an "intelligent" data density reduction algorithm. The algorithm presented here is intelligent in some respects, but not in others. On the one hand, it does not randomly remove points without considering their value to the surface being described. However, it is not designed to adapt to different levels of surface (terrain) variability. It would be desirable to have a truly intelligent program that could determine the ideal radius and tolerance values which should be used for reducing the density of a particular portion of the data set.

This enhancement to the thinning process could be accomplished by incorporating a two-dimensional Fast Fourier Transform (FFT) routine into the program. By subdividing the area of interest, and determining the frequency content of the harmonics represented in a patch of data, one could establish whether that region was highly variable (high frequency) or relatively flat and unchanging in depth (low frequency). As a result, an appropriate radius value could be chosen, based on the effective sampling frequency for that level of variability. With an approach similar to adaptive quadrature, the subdivisions of more excessively variable terrain could be further divided so that the best radius parameter would be selected to capture the topographic characteristics of that area. Although this FFT analysis would help in selecting an appropriate neighborhood for the thinning process, the user would still be left with choosing an acceptable tolerance value.

Using the FFT Analysis tool in the Imagine application, each of the three broadly defined test sites (Flats, Jet-Chan, and Reefs) was examined for frequency content. First, each data set was imported and converted into a gridded surface. The cell size used for the Flats and Jet-Chan data defaulted to 3 ft by 3 ft, and for the Reefs data, it was set at a 4 ft by 4 ft grid spacing. Once the data were gridded, the FFT tool could be used to generate Fourier magnitude images. Each image consisted of a number of rows and columns fixed at 2 to a power of “N.” (By design, the FFT process rounds-up, or pads, the rows and columns of the input surface to the closest 2^N value.) Near the center, but offset from the horizontal axis, a representative horizontal profile sample was taken across each of the three Fourier images. The result was a graph of harmonics (frequency components) versus magnitude (amount of frequency) configured in the frequency domain. From these profiles an optimal radius value was calculated. (Figures 14, 15, and 16 in the Results and Discussion chapter show profiles for each data set.)

Using the profile, an approximate “cut-off” frequency (the point where the signal appears to end) was determined. The number of cells along the horizontal profile represents the range of frequency in the signal. This value is relative to the signal width, the width of the padded input surface generated from the source data (Press, Teukolsky, Vetterling, and Flannery 1992). In each case, the raw data points were gridded into a surface of rows and columns. The cells making up these rows and columns could be any number of units in dimension. For instance, if the cell was not one foot by one foot, but rather three feet by three feet, the calculations would take this into consideration by multiplying the signal width by three. The cut-off frequency over the adjusted signal width would then provide the number of cycles per signal width, and, ultimately, the number of distance units per cycle (wavelength) could be derived. In accordance with the Nyquist Theory, there must be at least two samples taken per cycle to capture a signal frequency. Therefore, the wavelength of the cut-off frequency must be divided by two to determine the optimum radius, or sampling interval.

RESULTS AND DISCUSSION

Effectiveness of Data Density Reduction

The results of the statistical comparison tests are summarized in Tables 1, 2, and 3. For the parameter pairs tested, the DDRA compared favorably to both the Systematic and Random thinning methods. In all but two cases, the Systematic test statistic is greater than one. This means that the null hypothesis (Systematic method is as effective as the DDRA) may be rejected for 98% of the test runs. The two cases where the null hypothesis could not be rejected occurred in the Reefs Lower Left and Lower Right Quads. Here, the parameters of r = 30 and t =0.3 resulted in removal of 54 % of the original data, and allowed the Systematic method to out-perform the DDRA. This was a notably different result than that seen in the Flats and Jet-Chan data sets where removal of 50-70 % of the data, nonetheless, resulted in rejection of the null hypothesis.

In all test cases, a general correspondence was found between percentage of data removed and RMS. As expected, the larger the tolerance value used, the more data points were removed by the algorithm. Similarly, the larger the radius value chosen, the more data were removed.

As for the Random method, it proved to be a consistently less adequate means of data reduction than the DDR algorithm. It was tested in fewer trial runs since the noted pattern of

Radius	Tolerance	Percent Removed	RMS_S	RMS_D	RMS_R	RMS_S / RMS_D	RMS_R / RMS_D	Chi-Square Threshold RMS_T
Flats Upper Left Quad (5,376 points, N=872,904)								
30	0.3	44	0.6365	0.5961	0.7578	1.07	1.27	0.4994
30	0.2	34	0.5324	0.4889		1.09		0.4994 *
30	0.1	18	0.3957	0.3178		1.25		0.4994 *
20	0.3	30	0.4918	0.4902		1.17		0.4994 *
20	0.2	22	0.4375	0.3429		1.28		0.4994 *
20	0.1	11	0.3074	0.2275		1.35		0.4994 *
10	0.3	7	0.2749	0.1702		1.61		0.4994 *
10	0.2	5	0.1935	0.1276		1.52		0.4994 *
10	0.1	3	0.1367	0.0853	0.139	1.6	1.63	0.4994 *
Flats Upper Right Quad (5,505 points, N=884,260)								
30	0.3	57	0.5676	0.5171	0.674	1.1	1.3	0.4994
30	0.2	46	0.4644	0.4093		1.13		0.4994 *
30	0.1	25	0.3365	0.2626		1.28		0.4994 *
20	0.3	40	0.4385	0.3655		1.2		0.4994 *
20	0.2	30	0.3653	0.2888		1.26		0.4994 *
20	0.1	15	0.2737	0.1766		1.55		0.4994 *
10	0.3	9	0.1916	0.1108		1.73		0.4994 *
10	0.2	6	0.1743	0.0867		2.01		0.4994 *
10	0.1	3	0.1096	0.0545	0.1056	2.01	1.94	0.4994 *
Flats Lower Left Quad (6,349 points, N=873,887)								
30	0.3	53	0.5425	0.5186	0.6114	1.05	1.18	0.4994
30	0.2	42	0.4682	0.4466		1.05		0.4994 *
30	0.1	24	0.3483	0.3138		1.11		0.4994 *
20	0.3	37	0.4355	0.3814		1.14		0.4994 *
20	0.2	28	0.3726	0.3214		1.16		0.4994 *
20	0.1	15	0.2801	0.226		1.24		0.4994 *
10	0.3	10	0.2243	0.1553		1.44		0.4994 *
10	0.2	7	0.1901	0.1281		1.48		0.4994 *
10	0.1	4	0.1379	0.0894	0.1361	1.54	1.52	0.4994 *
Flats Lower Right Quad (6,236 points, N=878,802)								
30	0.3	63	0.5465	0.5171	0.6153	1.06	1.19	0.4994
30	0.2	52	0.4677	0.4204		1.11		0.4994 *
30	0.1	30	0.342	0.2649		1.29		0.4994 *
20	0.3	44	0.4204	0.3501		1.2		0.4994 *
20	0.2	34	0.362	0.2763		1.31		0.4994 *
20	0.1	18	0.2519	0.1772		1.42		0.4994 *
10	0.3	11	0.2219	0.1158		1.92		0.4994 *
10	0.2	8	0.1696	0.0905		1.87		0.4994 *
10	0.1	3	0.1126	0.0571	0.1197	1.97	2.1	0.4994 *

Table 1: Flats Data Summary (* indicates H_O is rejected).

Radius	Tolerance	Percent Removed	RMS_S	RMS_D	RMS_R	RMS_S / RMS_D	RMS_R / RMS_D	Chi-Square Threshold RMS_T
Jet-Chan Upper Left Quad (7,074 points, N=517,806)								
30	0.3	65	1.6368	0.9772	2.2629	1.67	2.32	0.4992
30	0.2	57	1.6422	0.805		2.04		0.4992
30	0.1	44	1.2448	0.6033		2.06		0.4992
20	0.3	51	1.4356	0.7262		1.98		0.4992
20	0.2	43	1.256	0.5941		2.11		0.4992
20	0.1	31	1.0404	0.4468		2.33		0.4992 *
10	0.3	22	0.9442	0.3704		2.55		0.4992 *
10	0.2	17	0.8704	0.3268		2.65		0.4992 *
10	0.1	11	0.6479	0.2579	0.687	2.51	2.66	0.4992 *
Jet-Chan Upper Right Quad (8,988 points, N=522,248)								
30	0.3	67	1.8833	1.4227	1.9922	1.32	1.4	0.4992
30	0.2	59	1.6205	1.1407		1.42		0.4992
30	0.1	41	1.3027	0.7382		1.76		0.4992
20	0.3	54	1.4786	0.9801		1.51		0.4992
20	0.2	46	1.3366	0.7907		1.69		0.4992
20	0.1	29	1.0854	0.5314		2.04		0.4992
10	0.3	28	1.0491	0.5266		1.99		0.4992
10	0.2	21	1.0012	0.4176		2.4		0.4992 *
10	0.1	11	0.6689	0.2669	0.6357	2.51	2.38	0.4992 *
Jet-Chan Lower Left Quad (8,228 points, N=524,267)								
30	0.3	65	2.263	1.7894	2.6177	1.26	1.46	0.4992
30	0.2	57	1.8191	1.531		1.19		0.4992
30	0.1	38	1.3956	0.916		1.52		0.4992
20	0.3	52	1.7693	1.2452		1.42		0.4992
20	0.2	43	1.6029	0.9798		1.64		0.4992
20	0.1	26	1.1133	0.6073		1.83		0.4992
10	0.3	24	1.0127	0.6207		1.63		0.4992
10	0.2	18	0.9402	0.5019		1.87		0.4992
10	0.1	9	0.6502	0.3137	0.5811	2.07	1.85	0.4992 *
Jet-Chan Lower Right Quad (8,710 points, N=525,720)								
30	0.3	71	2.6316	1.8678	2.9026	1.41	1.55	0.4992
30	0.2	63	2.0607	1.4547		1.42		0.4992
30	0.1	43	1.5385	0.8541		1.8		0.4992
20	0.3	59	2.0411	1.2482		1.64		0.4992
20	0.2	49	1.5758	0.9544		1.65		0.4992
20	0.1	30	1.2314	0.572		2.15		0.4992
10	0.3	29	1.228	0.5505		2.23		0.4992
10	0.2	21	0.9437	0.4116		2.29		0.4992 *
10	0.1	10	0.6821	0.2505	0.8067	2.72	3.22	0.4992 *

Table 2: Jet-Chan Data Summary (* indicates H_O is rejected).

Radius	Tolerance	Percent Removed	RMS_S	RMS_D	RMS_R	RMS_S / RMS_D	RMS_R / RMS_D	Chi-Square Threshold RMS_T
Reefs Upper Left Quad (7,645 points, N=1,206,290)								
30	0.3	53	0.5815	0.5553	0.7126	1.05	1.28	0.4995
30	0.2	42	0.4665	0.4211		1.11		0.4995 *
30	0.1	29	0.3741	0.3212		1.16		0.4995 *
20	0.3	37	0.4479	0.3812		1.17		0.4995 *
20	0.2	27	0.3886	0.3035		1.28		0.4995 *
20	0.1	17	0.2974	0.2275		1.31		0.4995 *
10	0.3	7	0.182	0.1056		1.72		0.4995 *
10	0.2	5	0.1519	0.0817		1.86		0.4995 *
10	0.1	3	0.1318	0.0643	0.141	2.05	2.19	0.4995 *
Reefs Upper Right Quad (7,686 points, N=1,227,904)								
30	0.3	52	0.7456	0.7101	0.8066	1.05	1.14	0.4995
30	0.2	40	0.6003	0.5877		1.02		0.4995
30	0.1	27	0.4764	0.4252		1.12		0.4995 *
20	0.3	35	0.5409	0.4784		1.13		0.4995 *
20	0.2	26	0.4929	0.3932		1.25		0.4995 *
20	0.1	17	0.3677	0.2967		1.24		0.4995 *
10	0.3	6	0.2438	0.1374		1.77		0.4995 *
10	0.2	4	0.1948	0.1134		1.72		0.4995 *
10	0.1	3	0.1395	0.0769	0.1377	1.81	1.79	0.4995 *
Reefs Lower Left Quad (7,299 points, N=1,207,109)								
30	0.3	54	0.4985	0.5179	0.6382	0.96	1.23	0.4995
30	0.2	42	0.4296	0.3876		1.11		0.4995 *
30	0.1	29	0.348	0.2916		1.19		0.4995 *
20	0.3	37	0.3863	0.3481		1.11		0.4995 *
20	0.2	27	0.3411	0.2802		1.22		0.4995 *
20	0.1	17	0.2721	0.2066		1.32		0.4995 *
10	0.3	7	0.1551	0.0997		1.56		0.4995 *
10	0.2	5	0.1361	0.0793		1.72		0.4995 *
10	0.1	3	0.109	0.0568	0.1083	1.92	1.91	0.4995 *
Reefs Lower Right Quad (8,237 points, N=1,220,653)								
30	0.3	54	0.9642	0.9815	1.2131	0.98	1.24	0.4995
30	0.2	42	0.7662	0.7052		1.09		0.4995
30	0.1	29	0.6367	0.5308		1.2		0.4995
20	0.3	37	0.7364	0.6255		1.18		0.4995
20	0.2	26	0.5948	0.4734		1.26		0.4995 *
20	0.1	17	0.5191	0.3732		1.39		0.4995 *
10	0.3	8	0.3363	0.183		1.84		0.4995 *
10	0.2	5	0.2402	0.1361		1.77		0.4995 *
10	0.1	3	0.1523	0.1038	0.1913	1.47	1.84	0.4995 *

Table 3: Reefs Data Summary (* indicates H_O is rejected).

higher (greater than one) test statistic values throughout the Fisher hypothesis testing made further iterations unnecessary. Because the method is random, it is possible that the removal of data by this approach could result in a more effectively thinned data set than one achieved through use of the DDR algorithm. However, that outcome is considered unlikely.

As discussed earlier, that Systematic and Random data density reduction methods are not very practical. It was predicted that the DDRA would produce more effective results than either of those methods. It is more important to note how well it did when compared to the measurement error in the data. The results of the chi-square test are listed in the final column of Tables 1, 2, and 3. The threshold RMS values with asterisks are those which call for rejection of the null hypothesis (the DDRA exceeds the system measurement error). With the Flats data, it can be seen that the algorithm produced RMS values under the threshold, as long as about half of the original points were retained. RMS values for the Reefs data were also generally under the threshold, but seemed to require that a slightly higher percentage of survey points be kept. The Jet-Chan, however, did not respond as well to the DDRA. In fact, only 25% of the data runs for that region resulted in rejection of the null hypothesis. Beyond about 30% data removal, the RMS values for the algorithm tended to exceed 0.5 ft. This is due to the higher frequency content of the Jet-Chan data set (as will be shown in the following section).

In each of the data runs, a standard deviation was computed for comparison with the RMS. It was found that the standard deviation and the RMS were generally the same to within at least three significant figures. This indicates that there were no obvious biases in the data density reduction algorithm.

Fourier Derived Radius Values

The fluctuations in a section of discretely sampled terrain can be mathematically modeled by sine and cosine functions in three dimensions. Each cross-section of that terrain could be described as a complicated wave which, no matter how irregular, can be modeled by a combination of sine and cosine waves of various frequencies and amplitudes. (See Appendix B for further details on the concepts and equations of Fourier analysis.)

Using a two-dimensional FFT, the individual components of a complex wave can be computed. The component sine and cosine waves differ in frequency and in their amount of contribution to the complicated wave. Added together, the component waves give a representation of a cross-section of terrain. If a section of terrain is primarily made up of waves in the low harmonics (low frequencies), then a larger radius value should suffice in the data density reduction algorithm. This is because terrain comprised of lower frequencies is relatively flat, or without a lot of variation in the elevation values over short horizontal distances. In this case, a large radius would be more efficient and could probably be used without significant loss of detail in the reduced DTM generated.

If the terrain is found to contain large quantities of high frequency component waves, it is more irregular in nature. Only by using a smaller radius value could one be sure to capture the detail of such variable terrain. The DDR algorithm works more slowly with a smaller radius parameter input, but tends to generate a more statistically accurate DTM.

The magnitude images generated by running an FFT process on the three data sets are pictured in Figures 11, 12, and 13. All of these images have the low frequencies concentrated

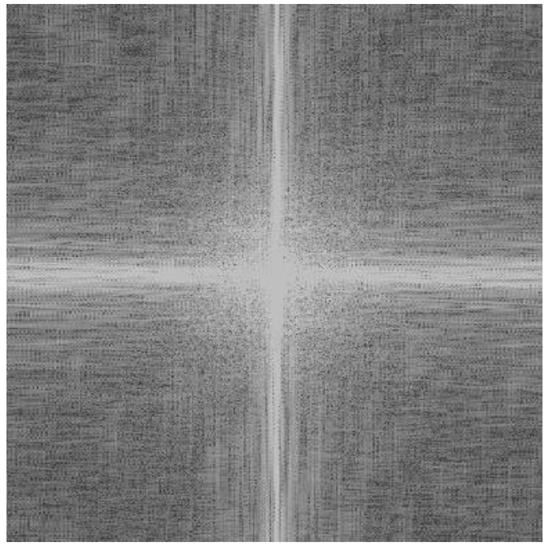

Figure 11. Flats FFT image

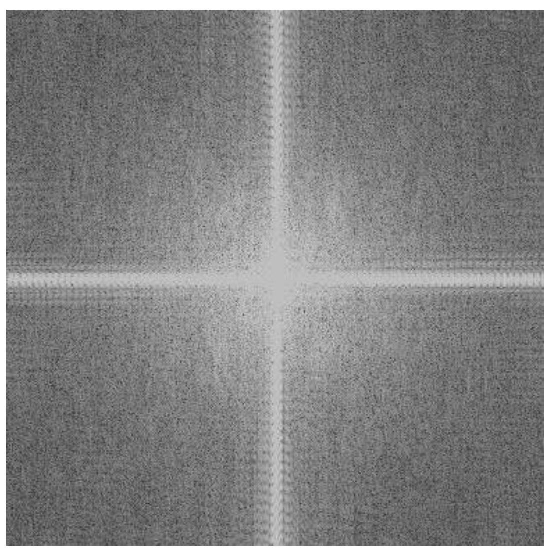

Figure 12: Jet-Chan FFT image

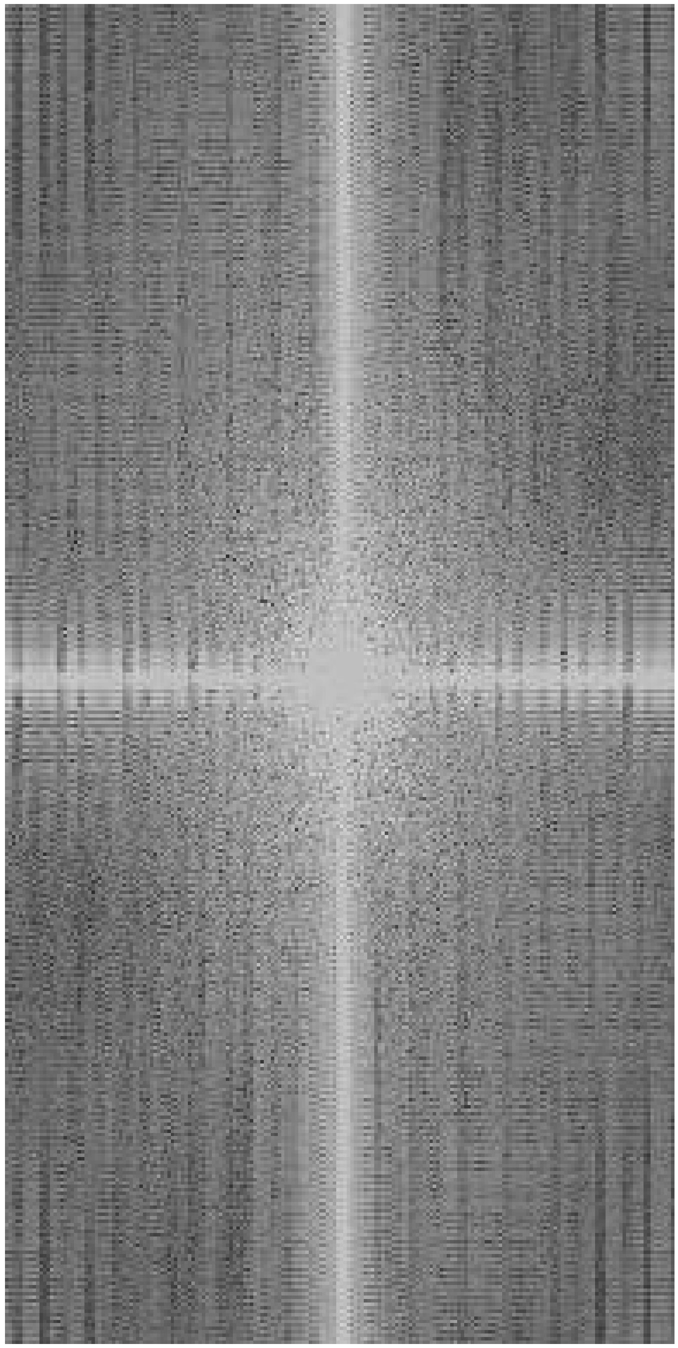

Figure 13: Reefs FFT image

at the center of the figure and the higher frequencies along the edges. Figures 14, 15, and 16 depict frequency profiles taken across the widths of the images. In each case, a peak can be seen in the graph. This peak envelops the frequency components (or harmonics) where the bulk of the information content (signal) in the terrain is concentrated. At its ends, each graph slopes downward into a region of “noise”. It is difficult at best to resolve exactly where the signal magnitude ends. By continuing the curve around the signal peak and down to the frequency axis, a cut of frequency can be determined (see Figure 14). This value is subjective and depends on how one interprets the profile (Press, Teukolsky, Vetterling, and Flannery 1992).

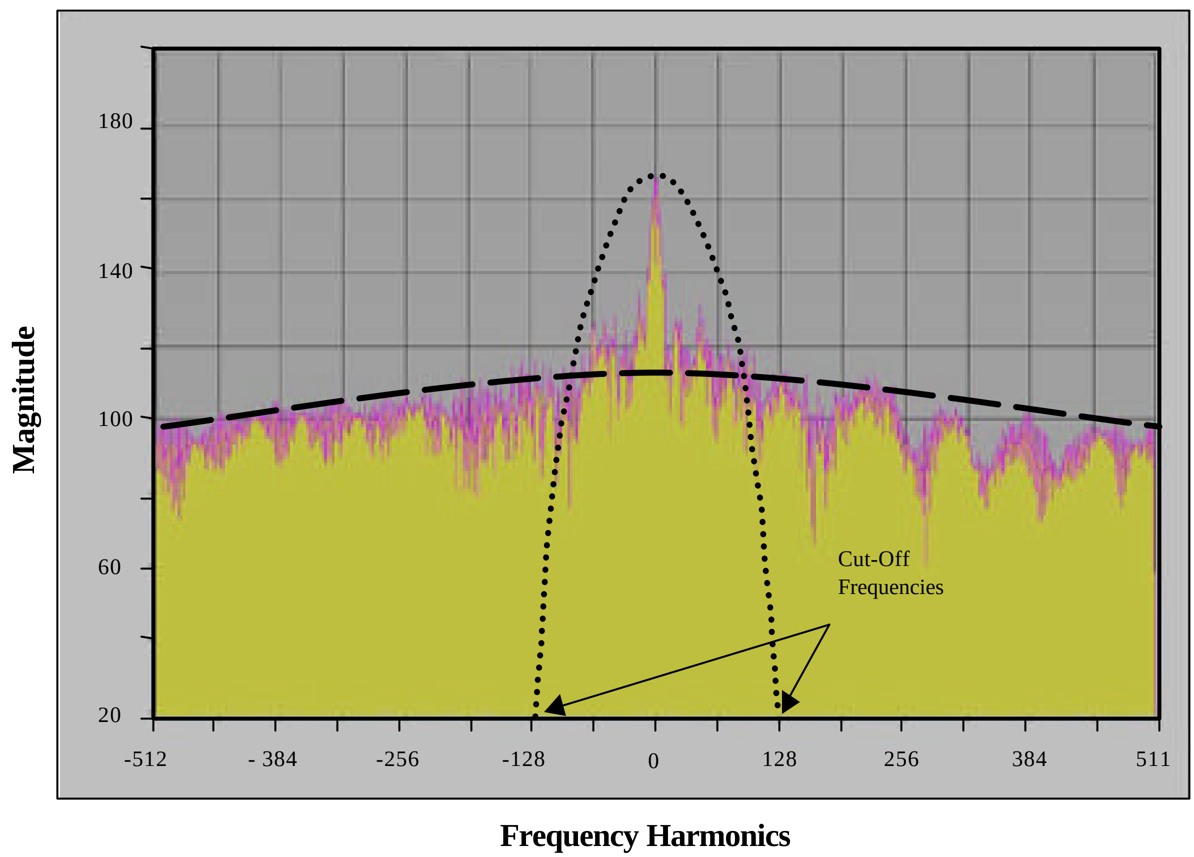

Figure 14: Flats FFT magnitude profile at row 472

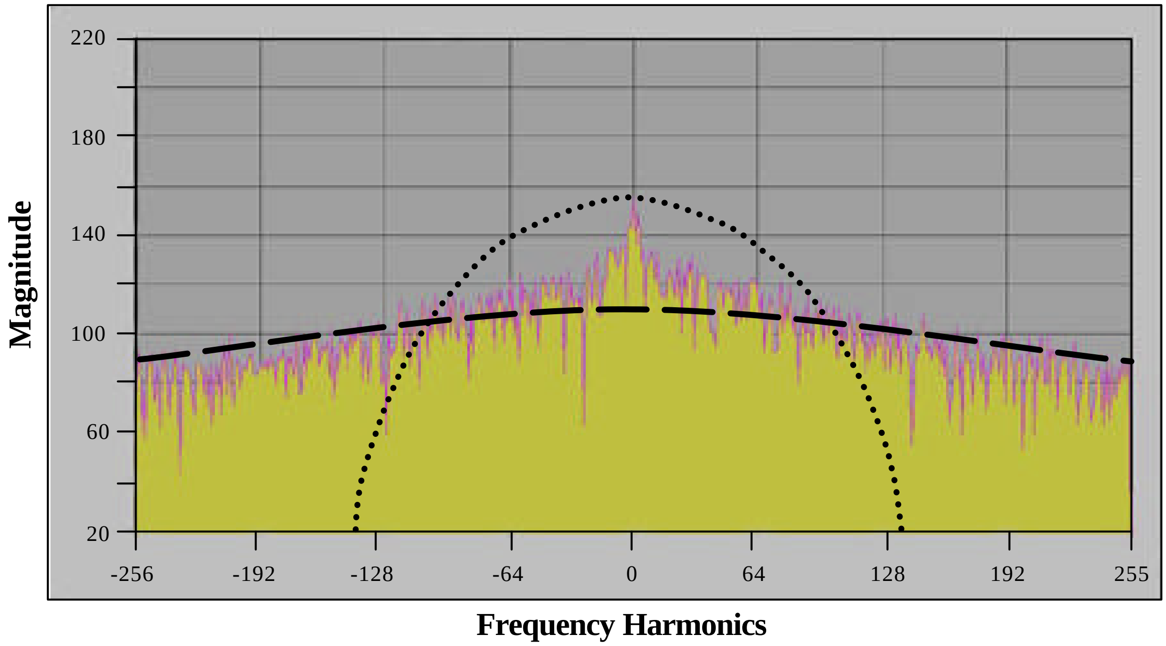

Figure 15: Jet-Chan FFT magnitude profile at row 226

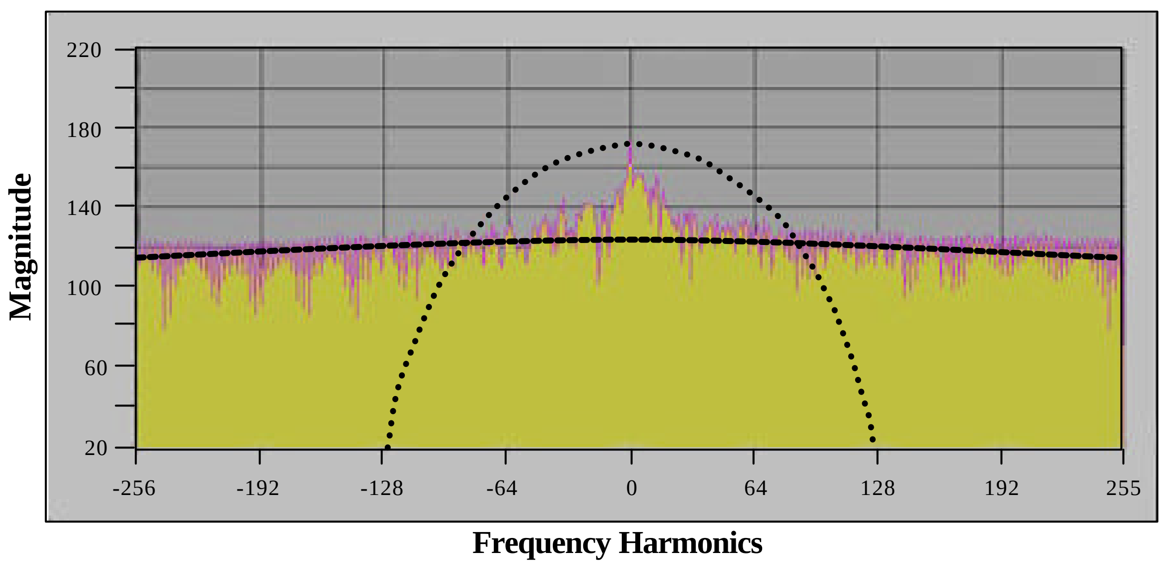

Figure 16: Reefs FFT magnitude profile at row 226

The optimum sampling frequency for a particular section of terrain is computed using this value found along the profile's x-axis. The cut-off frequency is divided by the signal width to determine the wavelength corresponding to this highest frequency. Then, as suggested by the Nyquist sampling rate, the radius value is selected at half that wavelength.

For Flats, the cut-off frequency was approximately 128 cycles and the signal width was 3072 feet. Therefore, the corresponding wavelength was 24 feet and the resulting radius value, 12 feet. Jet-Chan had a cut-off frequency of about 135 cycles with a signal width of 1536 feet, and, hence, a wavelength and radius of 11 and 6 feet, respectively. Reefs had a cut-off close to 128 cycles, and, with its signal width of 2048 feet, gave a wavelength and radius of 16 and 8 feet, respectively.

Based on the above calculations, the Jet-Chan data require a smaller radius than the other two data sets. This corresponds with the results of the chi-square tests on the RMS values. One would expect to have to sample the Flats area less frequently than the other two areas having higher frequency content. The sharp vertical slopes of the Jet-Chan and the more variable terrain of the Reefs should call for a smaller radius. All three computed radii are small compared to the ranges tested (i.e. 10, 20, and 30 feet). This implies that some actual signal content is being lost in the decimation; however, this loss is within the random noise of the system.

SUMMARY

Conclusions

This research focused on the development of an algorithm that would reduce the density of a lidar survey without affecting the coordinates of the surveyed data points. It was important to first establish what types of parameters affect intelligent data density reduction. Horizontal and vertical values termed the "radius" and the "tolerance," respectively, were identified as the two key input parameters.

The next step was to determine reasonable values for these parameters. In order to accomplish this, the parameters were related back to the data. The radius was associated with sampling frequency, and the tolerance with vertical measurement error. An effort to use Fast Fourier Transforms to identify optimum radius values based on optimum sampling frequency could then be made. Additionally, by keeping tolerance value selections below the value of stated vertical error in the data, less accuracy was sacrificed to the thinning process.

Finally, a method of determining how well a reduced DTM represents the surveyed terrain data was devised. The technique involved fitting a surface to the original data, and another to the thinned data set in order to establish the difference between the two surfaces (RMS). Using statistics, the results of the DDR algorithm were compared to the vertical measurement error of the data and to the results achieved by systematic and random thinning processes.

Based on data test runs, it was noted that, whenever a data set is thinned, at least some of its information content is lost. The goal is to minimize that loss of signal while maximizing the amount of data reduction achieved. As expected, it was found that the algorithm was significantly more effective than the systematic and random methods of thinning. It was also shown that the DDRA could thin data while remaining within the stated measurement error of the original survey. Over sixty percent of the test runs resulted in rejection of the hypothesis that the DDRA exceeds system measurement error. The data set with the highest spatial frequencies yielded the worst results in using the DDRA. Only about 10-20% of the data in that region could be removed before the DDRA RMS began to exceed the threshold.

A Fourier analysis technique was used in an effort to select an optimum radius value based on the frequency content of the data being thinned. The radii values calculated for the three test areas were considered reasonable, but uncertain. There is a significant amount of uncertainty in the procedure which involves the interpretation of a profile of FFT magnitude values. The determination of where the signal ends and where the noise begins is rather subjective, and will influence the calculated results.

Recommendations

Several recommendations can be made in light of this research. First, it would be beneficial to further test the Fourier analysis technique on areas of known terrain variability. A more concrete method of interpreting of the FFT image profiles would add consistency to the approach.

If the Fourier analysis were performed on localized areas of the surveyed terrain, appropriate radii values could be chosen for specific regions. Using a method analogous to adaptive quadrature, areas of greater frequency content could be further sub-divided until an appropriate radius was found. The choice of various radii based on terrain variability would enable the algorithm to work more effectively and efficiently. Ultimately, the FFT analysis procedure should be incorporated into the thinning program rather than being used separately.

Another program improvement involves the manner in which the user selects the parameters for the thinning algorithm. It would be ideal if the user could specify the desired accuracy (RMS) of the thinned data set and have the program find the appropriate parameters to achieve that result. Incorporation of the Fourier technique might help in the choice of effective radius values. It was considered that Fourier might also assist in determining a tolerance in the vertical dimension. There was difficulty noted in that many software packages add contrast stretch to the output data and shield the true values from the user. This process can be detrimental when attempting to analyze the raw data.

It was mentioned earlier that the gridding may be a viable method of thinning, especially if done within the horizontal and vertical measurement error of the data. The choice of thinning technique should consider the effectiveness, cost, and speed of the process and also the ultimate application for the data. It would be interesting to test the DDRA against various gridding methods to determine just how much positional accuracy is lost. If the loss is negligible, when compared to the total measurement error in the lidar survey, then gridding might be further considered as an effective way of reducing the data.

Other methods might be tested for use in determining how well a thinned data set represents surveyed terrain data. The method of choice in the DDRA was a surface generation technique involving IDW interpolation. Different types of interpolation could be used, or different methods altogether, such as the comparison of two Triangulated Irregular Networks (TINs).

Methods for improving the DDR algorithm have been and should continue to be considered. One such refinement involves a modification to the "marking" of redundant records for removal from the set. The idea is that an operating point (OP) which is used in determining that a test point (TP) within its neighborhood should be marked for removal, could be "preserved." Preserved means that the OP would be given a save-flag (in the positive position) which would ensure that, regardless of further analysis and tolerance checks, the OP would be retained in the final thinned data set. The positive save-flag would indicate that, since one or more test points were removed based on the location of the operating point, the operating point must remain, whether redundant (when compared to other neighborhood points) or not. This version of the algorithm would also be modified such that only test points could be marked for removal. Since every OP is eventually considered as a TP for points in its neighborhood, this change would not be a problem. While this modification to the algorithm would likely enhance accuracy, it would come at the sacrifice of thinning percentage.

Finally, with respect to the operating speed of the algorithm, another improvement could be made. The entire algorithm could be programmed in C or some other compiled language.

The ArcView application is useful in seeing the effects of the algorithm, but the interface no doubt slows the processing time considerably.

APPENDIX A
AVENUE CODE

```
*******************************************
'Name:  Proj.about_app
'Author/Date:  Pam LaFontaine / 26 April 98
'Updated_by/Date:  8 Feb 99
'Course:          URP6272, Spring 1998
'Prepared at:     GeoPlan Center, University of Florida
'---------------------------------------------------
'Description:  Shows a MsgBox with info about the SHOALS Application
'Requires:
'Calls:
'Is called by:  (Menu request -- About Application)
'Self:
'Returns:
'FileName:  Proj.about_app
'---------------------------------------------------
MsgBox.Report("This application was created to allow the SHOALS team to 'thin down'"++
        "the very dense data sets they collect by laser bathymetry.  The user"++
        "may select an area of interest (within the data set), run the thinning"++
        "algorithm, and then calculate statistics to indicate just how much"++
        "accuracy was sacrificed by removing the 'redundant' data points.  The"++
        "'Add As Theme' option allows the user to quickly create a new analysis"++
        "theme from an area within the data set."+NL+"A single shapefile theme"++
        "must be active to access most of the menu options.",
        "About the SHOALS Application")
*******************************************

*******************************************
'Name:  Proj.add_theme
'Author/Date:  Pam LaFontaine / 24 April 98
'Updated_by/Date:  22 Jan 99
'Course:          URP6272, Spring 1998
'Prepared at:     GeoPlan Center, University of Florida
'---------------------------------------------------
'Description:  Creates a shapefile from theBitmap (selected new data set values)
'         and adds the new theme to the view.
```

```
'Requires:  One active shapefile theme
'Calls:
'Is called by:  Proj.make_area_1s and Proj.fast_make_area_1s and Proj.run_many_thins
'Self:  theView = SELF.Get(0), t = SELF.Get(1)
'Returns:
'FileName:  Proj.add_theme
'---------------------------------------------------
'  Get the active view and the active theme
theView = SELF.Get(0)
' t is the same as "theTheme"
t = SELF.Get(1)
' ***************************************
if (t.Is( FTHEME ).Not) then
   if (t.CanExportToFtab.Not) then
      return NIL
   end
end

if (t.Is (FTHEME).Not) then
   def = av.GetProject.MakeFileName("theme", "shp")
   'def = FileDialog.Put(def, "*.shp", "Convert " + t.getName)
   if (def = NIL) then
      return NIL
   end
   anFTab = t.ExportToFtab(def)
else
   tbl = t.GetFTab
   attribVis = FALSE
   for each f in tbl.GetFields
      if ((f.IsVisible) and not (f.IsTypeShape)) then
         attribVis = TRUE
         break
      end
   end

   shapeVis = tbl.FindField("Shape").IsVisible
   if ((attribVis and shapeVis).Not) then
      return NIL
   end

   def = av.GetProject.MakeFileName("theme", "shp")
   'def = FileDialog.Put(def, "*.shp", "Convert " + t.getName)
   if (def = NIL) then
```

```
      return NIL
    end

    shpfld = (tbl.FindField("Shape"))
    if (shpfld.IsVisible.Not) then
      shpfld.SetVisible(shpfld.IsVisible.Not)
      WasNotVisible = TRUE
    else
      WasNotVisible =FALSE
    end

    anFTab = tbl.Export(def, Shape, tbl.GetSelection.Count > 0)
    if (anFTab.HasError) then
      if (anFTab.HasLockError) then
        MsgBox.Error("Unable to acquire Write Lock for file " + def.GetBaseName,"")
      else
        MsgBox.Error("Unable to create " + def.GetBaseName,"")
      end
      return NIL
    end

    if (WasNotVisible) then
      shpfld.SetVisible(FALSE)
    end
end

'create a theme and add it to the View
fthm = FTheme.Make(anFTab)
theView.AddTheme(fthm)
'bring the View to the front
theView.GetWin.Activate
*******************************************

*******************************************
'Name:  Proj.comp_table
'Author/Date:  Pam LaFontaine / 20 May 98
'Updated_by/Date:  20 April 99
'Course:           URP6272, Spring 1998
'Prepared at:      GeoPlan Center, University of Florida
'---------------------------------------------------
'Description:  Computes two comparison tables for multiple parameter combinations
'              of radius and tolerance values.  One table shows what percentage of the
'              original data set is represented by the new (thinned) data set.
```

```
'            The other shows the RMS (Root Mean Squared) value resulting from
'            the thinning process in each case.
'Requires:
'Calls:  Proj.fast_make_area_1s and Proj.fast_tab_stats
'Is called by:  (Menu request -- Add Two Dimensional Table)
'Self:
'Returns:
'FileName:  Proj.comp_table
'--------------------------------------------------

'  Set flag to direct program to proper subroutines for make theme (Will not add the
'  new thinned data sets to the view...there are too many!  Use "Run multiple thins" to
'  accomplish this.)
_compFlag = true

'  Get the active view and the active theme
theView = av.GetActiveDoc
theTheme = theView.GetActiveThemes.Get(0)

'  Allows you to go back out to the menu window to set the Analysis Properties to
'  match the Area of Interest!  If this is not done, your process will BOMB!
ans = MsgBox.YesNo("Have you set your Analysis Properties to correspond with the AOI?  ",
     "First Set Analysis Properties", true)
if (ans = false) then
   exit
end

'  Add the selected portion of the data set to the view as a theme
if (_portion) then
   MsgBox.Info("Since you've selected only a portion of the original data set, that portion"++
            "will be added to the view as a theme.","")
   av.Run("Proj.add_theme", {theView, theTheme})
   _portion = false
end

'  If a portion of a theme was selected, it will now be the top most theme in the
'  list of themes  It will become the "working theme"!
'  Select the working theme and the working VTab for the area selected
theTheme = theView.GetActiveThemes.Get(0)
theVTab = theTheme.GetFTab

'  Find the "area_pts" fields in the table
_areaField = theVTab.FindField("area_pts")
```

```
theBitmap = theVTab.GetSelection

theSelection = theBitmap.count
'  If user has failed to select an area first, exit!
if ((theSelection = NIL) OR (theSelection <= 0)) then
   MsgBox.Info("Must select an area to be thinned first...use Select Area option.", "")
   exit
end

'  Give a value of "1" to "area_pts" field of records in the selected area
if (theVTab.StartEditingWithRecovery) then
   for each rec in theBitmap
      theVTab.BeginTransaction
      theVTab.SetValue(_areaField, rec, 1)
      theVTab.EndTransaction
   end
else
  MsgBox.Warning("Problem Editing Table","")
  exit
end

'MsgBox.Info("The number of selected points is "+theBitmap.Count.AsString,"")
originalRecValue = theBitmap.Count

' ---------------------------------------------------------------

MsgBox.Info("By entering the number of tolerance and radius values, you will determine the"++
         "dimensions of two 't' by 'r' matrices.", "")

numRows = MsgBox.Input("How many tolerance values do you wish to try?",
                 "Number of Rows", "")

if ((numRows = nil) or (numRows.IsNumber = false)) then
   exit
end
if (numRows.AsNumber < 1) then
  exit
end
' ---------------------------------------------------------------
numCols = MsgBox.Input("How many radius values do you wish to try?",
                 "Number of Columns", "")
```

```
if ((numCols = nil) or (numCols.IsNumber = false)) then
    exit
end
if (numCols.AsNumber < 1) then
   exit
end
' ------------------------------------------------------------

tolLabelList={}
for each tNum in 1..numRows.AsNumber
    tolString = "tolerance #"+tNum.AsString+":  "
    tolLabelList.Add(tolString)
end
tolDefaultList = {}

radLabelList={}
for each rNum in 1..numCols.AsNumber
    radString = "radius #"+rNum.AsString+":  "
    radLabelList.Add(radString)
end
radDefaultList = {}

tolStringList = {}
tolStringList = MsgBox.MultiInput("Enter tolerance values you wish to try (in feet):  ",
                        "Enter Tolerance Data", tolLabelList, tolDefaultList)

tsIterations = tolStringList.count-1
tolList ={}
for each stringVal in 0..tsIterations
    tolString = tolStringList.Get(stringVal)
    tolList.Add(tolString.AsNumber)
end

' ************** NOW tolList is a list of NUMBERS! ***************

radStringList = {}
radStringList = MsgBox.MultiInput("Enter the radius values you wish to try (in feet):  ",
                        "Enter Radius Data", radLabelList, radDefaultList)

rsIterations = radStringList.count-1
radList ={}
for each stringVal in 0..rsIterations
    radString = radStringList.Get(stringVal)
```

```
    radList.Add(radString.AsNumber)
end

' ************** NOW radList is a list of NUMBERS! ***************

rIterations = numRows.AsNumber-1
for each tIndex in 0..rIterations
    if ((tolList.Get(tIndex) = nil) or (tolList.Get(tIndex) < 0)) then
        MsgBox.Error("You have input an invalid tolerance!","")
        exit
    end
end

cIterations = numCols.AsNumber-1
for each rIndex in 0..cIterations
    if ((radList.Get(rIndex) = nil) or (radList.Get(rIndex) < 0)) then
        MsgBox.Error("You have input an invalid radius!","")
        exit
    end
end

' --------------------------------------------------------------

fieldList = List.Make
tolField = Field.Make("tol", #FIELD_FLOAT, 8, 4)
fieldList.Add(tolField)
for each numEl in 0..cIterations
    theNum = radList.Get(numEl)
    theName = "r"+theNum.AsString
    theField = Field.Make(theName, #FIELD_FLOAT, 8, 4)
    fieldList.Add(theField)
end
'---------------------------------------------------------------
' Create a dBase table to hold standard deviation results of comparison runs

RMSFileName = av.GetProject.GetWorkDir.MakeTmp("RMS", "dbf")
compVTab = VTab.MakeNew(RMSFileName, dBase)
if (compVtab.IsEditable) then
    compVTab.AddFields(fieldList)
    compVTab.SetEditable(false)
else
    Msgbox.Info("Problem editing compVTab!","")
    exit
```

```
end

compVTab = VTab.Make(RMSFileName,false,false)
compTable = Table.Make(compVTab)
compTable.SetName(RMSFileName.GetBaseName)
'compTable.GetWin.Open

'-------------------------------------------------------------
'  Create a dBase table to hold the percent thinned information

thinFileName = av.GetProject.GetWorkDir.MakeTmp("pcent","dbf")
thinVTab = VTab.MakeNew (thinFileName, dBase)
if (thinVtab.IsEditable) then
   thinVTab.AddFields(fieldList)
   thinVTab.SetEditable(false)
else
   Msgbox.Info("Problem editing thinVTab!","")
   exit
end

thinVTab = VTab.Make(thinFileName,false,false)
thinTable = Table.Make(thinVTab)
thinTable.SetName(thinFileName.GetBaseName)
'thinTable.GetWin.Open

'-------------------------------------------------------------
compVTab.StartEditingWithRecovery
thinVTab.StartEditingWithRecovery
compVTab.BeginTransaction
thinVTab.BeginTransaction
'-------------------------------------------------------------
'  rIterations = numRows.AsNumber-1 (defined above)
for each tolEl in 0..rIterations
   tolValue = tolList.Get(tolEl) 'IT IS A NUMBER!
   '-------------------------
   cRec = compVTab.AddRecord
   tolField = compVTab.FindField("tol")
   compVTab.SetValue(tolField, tolEl, tolValue)
   '-------------------------
   tRec = thinVTab.AddRecord
   tolField = thinVTab.FindField("tol")
   thinVTab.SetValue(tolField, tolEl, tolValue)
   '-------------------------
```

```
   ' cIterations = numCols.AsNumber-1 (defined above)
   for each radEl in 0..cIterations
      radValue = radList.Get(radEl) 'IT IS A NUMBER!
      theFieldName = "r" + radValue.AsString
      compField = compVTab.FindField(theFieldName)
      thinField = thinVTab.FindField(theFieldName)
      '--------------------------
      '  Send radius and tolerance value to subroutines to thin data
      fthm = av.Run("Proj.fast_make_area_1s", {theView, theTheme, theVTab, radValue,
tolValue})
      '--------------------------
      denseValue = originalRecValue
      thinValue = _newRecValue
      percent = (thinValue/denseValue) * 100
      '--------------------------
      '  Send fthm to subroutines to calculate Root Mean Squared (RMS)
      RMSsurf = av.Run("Proj.fast_tab_stats", {theTheme, fthm})
      '--------------------------
      compVTab.SetValue(compField, tolEl, RMSsurf)
      thinVTab.SetValue(thinField, tolEl, percent)
   end

   '_pDict.Add(tolEl.AsString, pList)
end
msgbox.info("Processing Complete","")
compVTab.EndTransaction
thinVTab.EndTransaction
compVtab.StopEditingWithRecovery(true)
thinVtab.StopEditingWithRecovery(true)

theVTab.StopEditingWithRecovery(false)
******************************************

******************************************
'Name:  Proj.convert
'Author/Date:  Pam LaFontaine / 24 April 98
'Updated_by/Date:  8 Feb 99
'Course:          URP6272, Spring 1998
'Prepared at:      GeoPlan Center, University of Florida
'--------------------------------------------------
'Description:  Creates a shapefile from theBitmap (selected new data set values)
'Requires:  One active shapefile theme
'Calls:
```

```
'Is called by:  (Menu request -- Add As Theme)
'Self:
'Returns:
'FileName:  Proj.convert
'---------------------------------------------------

'  Get the active view and the active theme
theView = av.GetActiveDoc
t = theView.GetActiveThemes.Get(0)

'  Ensure user has the view and theme he wants for conversion.
' ****************************************
theViewName = theView.GetName
theThemeName = t.GetName

if ((t = nil) OR (theView = nil)) then
   MsgBox.Error("You must have one view and one theme active to run this function!","")
   exit
else
   ans = MsgBox.YesNo("Are you creating a shapefile from the "+theThemeName.AsString+"
theme in the "+
        theView.AsString+" view?","Are you sure about input?", true)
   if (ans = false) then
      exit
   end
end
' *****************************************

if (t.Is( FTHEME ).Not) then
   if (t.CanExportToFtab.Not) then
      return NIL
   end
end

if (t.Is (FTHEME).Not) then
   def = av.GetProject.MakeFileName("theme", "shp")
   def = FileDialog.Put(def, "*.shp", "Convert " + t.getName)
   if (def = NIL) then
      return NIL
   end
   anFTab = t.ExportToFtab(def)
else
   tbl = t.GetFTab
```

```
    attribVis = FALSE
    for each f in tbl.GetFields
      if ((f.IsVisible) and not (f.IsTypeShape)) then
        attribVis = TRUE
        break
      end
    end

    shapeVis = tbl.FindField("Shape").IsVisible
    if ((attribVis and shapeVis).Not) then
      return NIL
    end

    def = av.GetProject.MakeFileName("theme", "shp")
    def = FileDialog.Put(def, "*.shp", "Convert " + t.getName)
    if (def = NIL) then
      return NIL
    end

    shpfld = (tbl.FindField("Shape"))
    if (shpfld.IsVisible.Not) then
      shpfld.SetVisible(shpfld.IsVisible.Not)
      WasNotVisible = TRUE
    else
      WasNotVisible =FALSE
    end

    anFTab = tbl.Export(def, Shape, tbl.GetSelection.Count > 0)
    if (anFTab.HasError) then
      if (anFTab.HasLockError) then
        MsgBox.Error("Unable to acquire Write Lock for file " + def.GetBaseName,"")
      else
        MsgBox.Error("Unable to create " + def.GetBaseName,"")
      end
      return NIL
    end

    if (WasNotVisible) then
      shpfld.SetVisible(FALSE)
    end
  end

if (MsgBox.YesNo("Add shapefile as theme to the "+theView.AsString+" view?",
```

```
            "Add Shapefile to View?",true).Not) then
  exit
end

'create a theme and add it to the View
fthm = FTheme.Make(anFTab)
theView.AddTheme(fthm)
'bring the View to the front
theView.GetWin.Activate
******************************************

******************************************
'Name:  Proj.disable_shape
'Author/Date:  Pam LaFontaine / 24 April 98
'Updated_by/Date:  16 Jan 99
'Course:          URP6272, Spring 1998
'Prepared at:      GeoPlan Center, University of Florida
'--------------------------------------------------
'Description:  Updates properties for "Add As Theme", "Select Area" and
'              "Run Thin" menu items
'Requires:
'Calls:
'Is called by:  (Menu request updates -- Add As Theme, Select Area and Run Thin
'Self:
'Returns:
'FileName:  Proj.disable_shape
'--------------------------------------------------

'  Script to disable certain menu items if there is more than
'  one theme active or if the theme is not an Ftheme

theView = av.GetActiveDoc
activeList = theView.GetActiveThemes
activeCount = theView.GetActiveThemes.Count
SELF.SetEnabled ((activeCount = 1) AND
(activeList.Get(0).Is(Ftheme)))
******************************************

******************************************
'Name:  Proj.fast_thin
'Author/Date:  Pam LaFontaine / 16 May 98
'Updated_by/Date:  25 Aug 99
'Course:          URP6272, Spring 1998
```

```
'Prepared at:      GeoPlan Center, University of Florida
'----------------------------------------------------
'Description:  Gets the VTab associated with the view's active theme.
'              Goes through each record of the attribute table and
'              tests whether or not the record should be "marked".
'              Records are "marked" if the z-value of the operating
'              point (Zop) falls within the user-established tolerance
'              of the z-value of the test point (Ztest).
'Requires:
'Calls:
'Is called by:  Proj.fast_make_area_1s
'Self:  theVTab = SELF.Get(0), theTheme = SELF.Get(1), dist = SELF.Get(2), tolerance =
SELF.Get(3)
'Returns:  theVTab
'FileName:  Proj.fast_thin
'----------------------------------------------------

'  Assign the SELF variables
theVTab = SELF.Get(0)
theTheme = SELF.Get(1)
dist = SELF.Get(2)
tolerance = SELF.Get(3)

'  Find the "mark", "area_pts" and "Z" fields in the table
_markField = theVTab.FindField("mark")
_areaField = theVTab.FindField("area_pts")
_zField = theVTab.FindField("Z")

'  User can determine the radius of the circle used to select the points for analysis.  (This value
'  is "dist" above brought in from a table of values.)
'  User can determine the tolerance within which a z-value (depth) will be rejected.
'  Tolerance is the difference value between two depths which is determined
'  to make the least shoal of the two depths redundant.  The redundant depths are
'  then "marked" in a separate field and will not be included in the output
'  (This value is "tolerance" above brought in from a table of values.)

'  Go through each record in selected area and perform analysis
for each index in theVTab
   area = theVTab.ReturnValue(_areaField, index)
   if (area = 0) then
      continue
   else
      mark = theVTab.ReturnValue(_markField, index)
```

```
        if (mark = 1) then
          continue
        else
          if (theVTab.StartEditingWithRecovery) then
            theBitmap = theVTab.GetSelection
            theBitmap.Clearall
            theVTab.UpdateSelection
            '  Put the point itself into bitmap set
            theBitmap.Set(index)
            theVTab.UpdateSelection
            '  Add to bitmap set the points within radius

theTheme.SelectByTheme(theTheme,#FTAB_RELTYPE_ISWITHINDISTANCEOF,
                           dist,#VTAB_SELTYPE_OR)
            theVtab.UpdateSelection
            Zop = theVTab.ReturnValue(_zField, index)

            '**********************************************************
            '  Marking process is shoal-biased...the most shoal (or least deep)
            '  of the two points is kept in the new set!
            for each rec in theVTab.GetSelection
              '  Do not make the Z-value comparison with points already marked for removal
              mark1 = theVTab.ReturnValue(_markField, rec)
              if (mark1 = 1) then
                continue
              elseif (rec <> index) then
                Ztest = theVTab.ReturnValue(_zField, rec)
                diff = Zop-Ztest
                if (diff.Abs < tolerance) then
                  if (Zop < Ztest) then
                    theVTab.BeginTransaction
                    '  mark the operating value's Z-field
                    theVTab.SetValue(_markField, index, 1)
                    theVTab.EndTransaction
                  else
                    theVTab.BeginTransaction
                    '  mark the test value's Z-field
                    theVTab.SetValue(_markField, rec, 1)
                    theVTab.EndTransaction
                  end
                else
                  continue
                end
```

```
            else
               continue
            end
         end
         ' *********************************************************
      else
         MsgBox.Warning("Problem Editing Table!","")
         exit
      end
    end
  end
end

return theVTab
*******************************************

*******************************************
'Name:  Proj.fast_make_area_1s
'Author/Date:  Pam LaFontaine / 18 May 98
'Updated_by/Date:  22 Jan 99
'Course:          URP6272, Spring 1998
'Prepared at:     GeoPlan Center, University of Florida
'---------------------------------------------------
'Description:  ("area_pts" values for user's selected points are already changed to 1)
'              This script runs the Proj.fast_thin and the Proj.add_theme subroutines.
'              It returns the ftheme generated from the thinned data set.
'Requires:  One active shapefile theme
'Calls:  Proj.fast_thin and (Proj.add_theme or Proj.make_theme)
'Is called by:  Proj.run_many_thins or Proj.comp_table
'Self:  theView = SELF.Get(0), theTheme = SELF.Get(1), theVTab = SELF.Get(2),
'       radValue = SELF.Get(3), tolValue = SELF.Get(4)
'Returns:  fthm
'FileName:  Proj.fast_make_area_1s
'---------------------------------------------------

'  Get the active view and the active theme
theView = SELF.Get(0)
theTheme = SELF.Get(1)
theVTab = SELF.Get(2)
radValue = SELF.Get(3)
tolValue = SELF.Get(4)

if ((theTheme = nil) OR (theView = nil) or (theVTab = nil)) then
```

```
    MsgBox.Error("You must have one view and one theme active to run this function!","")
    exit
end

theBitmap = theVTab.GetSelection
theBitmap.Clearall
theVTab.UpdateSelection

theVTab = av.Run("Proj.fast_thin", {theVTab, theTheme, radValue, tolValue})

theBitmap = theVTab.GetSelection
theBitmap.Clearall
theVTab.UpdateSelection

'  Use queries to select only the new set of points
theQuery1 = "([area_pts] = 1)"
theVTab.Query(theQuery1, theBitmap, #VTAB_SELTYPE_NEW)
theVTab.UpdateSelection

theQuery2 = "([mark] = 0)"
theVTab.Query(theQuery2, theBitmap, #VTAB_SELTYPE_AND)
theVTab.UpdateSelection

'MsgBox.Info("The number of points in the new set is "+theBitmap.Count.AsString,"")
_newRecValue = theBitmap.Count

if (_compFlag = true) then
    '  Create a shapefile from the new data set, but do not add it to the view
    fthm = av.Run("Proj.make_theme", {theView, theTheme})
elseif (_compFlag = false) then
    '  Create a shapefile from the new data set and add it to the view
    av.Run("Proj.add_theme", {theView, theTheme})
    fthm = nil
else
    exit
end

theBitmap = theVTab.GetSelection
theBitmap.Clearall
theVTab.UpdateSelection

'  Ensure that the values in the "Mark" field are set to zero
'  If not, INITIALIZE!
```

```
'  The working values of the "area_pts" are left at "1" since we continue to use that
'  set for the remaining runs.
if (theVTab.StartEditingWithRecovery) then
   for each rec in theVTab
      if (theVTab.ReturnValue(_markField, rec)<>0) then
         theVTab.BeginTransaction
         theVTab.SetValue(_markField, rec, 0)
         theVTab.EndTransaction
      end
   end
else
   MsgBox.Warning("Problem Editing Table!","")
   exit
end

'  Return the newly created fthm to Proj.run_many_thins or to
'  Proj.comp_table to be used
return fthm
*******************************************

*******************************************
'Name:  Proj.fast_stats_calc
'Author/Date:  Pam LaFontaine / 16 May 98
'Updated_by/Date:  25 Aug 99
'Course:           URP6272, Spring 1998
'Prepared at:      GeoPlan Center, University of Florida
'---------------------------------------------------
'Description:  Performs grid operations and creates a statistics table
'Requires:
'Calls:
'Is called by:  Proj.fast_stats
'Self:  denseSurf = SELF.Get(0), sparseSurf = SELF.Get(1)
'Returns:  statsList = {statsSurfVTab, sumSurfVTab, numCells}
'FileName:  Proj.fast_stats_calc
'---------------------------------------------------

theView = av.GetActiveDoc

'  Identify all of the grids being passed from "Proj.fast_stats" Script
denseSurf = SELF.Get(0)
sparseSurf = SELF.Get(1)

'  *******************************************
```

```
' Make a Zone Grid (Integer Grid) to define which cells will be used in calculations
' In actuality all cells of each surface grid will be used (there should be no null
' values if the interpolation process was effective)
zoneGrid = (denseSurf.IsNull).Con(0.AsGrid, 1.AsGrid)

' ******************************************

' Subtract the sparse surface from the dense surface ...
' there is a difference value for each cell location
surfDiff = denseSurf - sparseSurf

' ******************************************

' Square the values that make up surfDiff (We will add these values together
' and divide by the total number of cells involved)
surfDiffSqr = surfDiff * surfDiff

' ******************************************

' Prj.MakeNull will be used for the zone projection to default to the
' current view's projection

' Identify the Zone Field for the zoneGrid
zoneGridTheme = GTheme.Make(zoneGrid)
zoneVTab = zoneGridTheme.GetVTab
if (zoneVTab = nil) then
  MsgBox.Info("Could not get the zoneVTab!","")
  exit
end
zoneField = zoneVTab.FindField("Value")

' (No Data = true) means that the statistics will not be computed for
' cells with a value of No Data (All of our cells should contain data anyway!)

' Create a filename for the zone statistics
zoneFN = av.GetProject.GetWorkDir.MakeTmp("zone","dbf")

' ******************************************

' Create the Zonal Statistics Table for surfDiff values
statsSurfVTab = (surfDiff).ZonalStatsTable(zoneGrid, Prj.MakeNull, zoneField, true, zoneFN)
```

```
' Check if problems creating the statsSurfVTab
if (statsSurfVTab.HasError) then
   MsgBox.Error("The statsSurfVTab has an error!","")
   exit
end

'
*************************************************************************

' Find the sum of the squares of "denseSurf - sparseSurf" (or surfDiff) residuals
sumSurfDiffSqr = (surfDiffSqr).ZonalStats(#GRID_STATYPE_SUM, zoneGrid,
Prj.MakeNull,
                                zoneField, true)

if (sumSurfDiffSqr.HasError) then
   MsgBox.Error("The sumSurfDiffSqr had an error!","")
   exit
end

' Retrieve the sum value stored in each cell, without loosing precision of the number
' (We will divide by 100 later to account for this step)
multSumSurfDiffSqr = (sumSurfDiffSqr * 100.AsGrid).Int

sumSurfVTab = multSumSurfDiffSqr.GetVTab
if (sumSurfVTab = nil) then
   MsgBox.Info("Could not get the sumSurfVTab!","")
   exit
end

'
*************************************************************************

' Find the number of cells in the surfDiffSqr grid.
' Find number of rows and number of columns and multiply
rcList = surfDiffSqr.GetNumRowsAndCols
numRows = rcList.Get(0)
' MsgBox.Info("The number of rows is "+numRows.AsString,"Rows")
numCols = rcList.Get(1)
' MsgBox.Info("The number of cols is "+numCols.AsString,"Cols")
numCells = numRows * numCols
' MsgBox.Info("The number of cells is "+numCells.AsString,"Total number of Cells")
```

```
'
***********************************************************************

statsList = {statsSurfVTab, sumSurfVTab, numCells }

return statsList
*******************************************

*******************************************
'Name:  Proj.fast_stats
'Author/Date:  Pam LaFontaine / 18 May 98
'Updated_by/Date:  9 Aug 99
'Course:           URP6272, Spring 1998
'Prepared at:      GeoPlan Center, University of Florida
'---------------------------------------------------
'Description:  Computes the various statistics to compare two point sets
'Requires:  Two shape themes
'Calls:  Proj.make_fast_grid, Proj.make_fast_surf, and Proj.fast_stats_calc
'Is called by:  Proj.run_many_thins
'Self:  run = SELF
'Returns:  valueList = {minimum, maximum, sd, RMSsurf}
'FileName:  Proj.fast_stats
'---------------------------------------------------

run = SELF

theView = av.GetActiveDoc
themesList = theView.GetThemes

'  Make sure that the only two active themes are the original selected set and the new thinned set
for each t in themesList
   t.SetActive(false)
end

themesList.Get(0).SetActive(true)
themesList.Get(run).SetActive(true)

activeThemes = theView.GetActiveThemes
activeThemeCount = theView.GetActiveThemes.Count

if (activeThemeCount = nil) then
   exit
elseif (activeThemeCount <> 2) then
```

```
   MsgBox.Info("You must have TWO shape themes selected to make this statictical"++
          "comparison", "")
   exit
elseif ((activeThemes.Get(0).Is(Ftheme).Not) OR (activeThemes.Get(1).Is(Ftheme).Not)) then
   MsgBox.Info("You must be working with shapefile themes and not grids or images!", "")
   exit
else
   theTheme1 = activeThemes.Get(0)
   theFTab1 = theTheme1.GetFTab
   theTheme2 = activeThemes.Get(1)
   theFTab2 = theTheme2.GetFTab
   FTab1count = theFTab1.GetNumRecords
   FTab2count = theFTab2.GetNumRecords
   if (FTab1count > FTab2count) then

      denseSurfaceGridThm = av.Run("Proj.make_fast_surf", theTheme1)
      denseSurfaceGrid = denseSurfaceGridThm.GetGrid

      sparseSurfaceGridThm = av.Run("Proj.make_fast_surf", theTheme2)
      sparseSurfaceGrid = sparseSurfaceGridThm.GetGrid
   else

      denseSurfaceGridThm = av.Run("Proj.make_fast_surf", theTheme2)
      denseSurfaceGrid = denseSurfaceGridThm.GetGrid

      sparseSurfaceGridThm = av.Run("Proj.make_fast_surf", theTheme1)
      sparseSurfaceGrid = sparseSurfaceGridThm.GetGrid
   end
end

surfList = {denseSurfaceGrid, sparseSurfaceGrid}

statsList = av.Run("Proj.fast_stats_calc", surfList)

statsSurfVTab = statsList.Get(0)
sumSurfVTab = statsList.Get(1)
numCells = statsList.Get(2)

'  Identify the "value" field of the individual cells in the sumSurfVTab grid
sumFieldSurf = sumSurfVTab.FindField("Value")

'  Retrieve the value which is in each of the cells of the sumSurfVTab grid
multSumSurfDiffSqr = sumSurfVTab.ReturnValue(sumFieldSurf, 0)
```

```
'  Divide multSumSurfDiffSqr by 100, since earlier (in Proj.fast_stats_calc) we multiplied
'  the number by this value in order to retain precision
sumSurfDiffSqr = multSumSurfDiffSqr/100

minimumField = statsSurfVTab.FindField("Min")
minimum = statsSurfVTab.ReturnValue(minimumField, 0)
maximumField = statsSurfVTab.FindField("Max")
maximum = statsSurfVTab.ReturnValue(maximumField, 0)
SDField = statsSurfVTab.FindField("Std")
sd = statsSurfVTab.ReturnValue(SDField, 0)

' ***********************************************************************

'  RMS (root mean square) is computed across all cells of the difference surface
RMSsurf = (sumSurfDiffSqr/numCells).Sqrt


valueList = {minimum, maximum, sd, RMSsurf}

return valueList
******************************************

******************************************
'Name:  Proj.fast_tab_stats_calc
'Author/Date:  Pam LaFontaine / 20 May 98
'Updated_by/Date:  25 Aug 99
'Course:         URP6272, Spring 1998
'Prepared at:     GeoPlan Center, University of Florida
'--------------------------------------------------
'Description:  Performs grid operations and creates a statistics table
'             One a table of "percentage thinned" the other a table
'             or Root Mean Square (RMS) values
'Requires:
'Calls:
'Is called by:  Proj.fast_tab_stats
'Self:  denseSurf = SELF.Get(0), sparseSurf = SELF.Get(1)
'Returns:  RMSsurf
'FileName:  Proj.fast_tab_stats_calc
'--------------------------------------------------

theView = av.GetActiveDoc
```

```
' Identify all of the grids being passed from "Proj.fast_tab_stats" Script
denseSurf = SELF.Get(0)
sparseSurf = SELF.Get(1)

' ******************************************

' Make a Zone Grid (Integer Grid) to define which cells will be used in calculations
' In actuallity all cells of each surface grid will be used (there should be no null
' values if the interpolation process was effective)
zoneGrid = (denseSurf.IsNull).Con(0.AsGrid, 1.AsGrid)

' ******************************************

' Subtract the sparse surface from the dense surface ...
' there is a difference value for each cell location
surfDiff = denseSurf - sparseSurf

' ******************************************

' Square the values that make up surfDiff (We will add these values together
' and divide by the total number of cells involved)
surfDiffSqr = surfDiff * surfDiff

' ******************************************

' Prj.MakeNull is used for the zone projection to default to the
' current view's projection

' Identify the Zone Field for the zoneGrid
zoneGridTheme = GTheme.Make(zoneGrid)
zoneVTab = zoneGridTheme.GetVTab
if (zoneVTab = nil) then
   MsgBox.Info("Could not get the zoneVTab!","")
   exit
end
zoneField = zoneVTab.FindField("Value")

' (No Data = true) means that the statistics will not be computed for
' cells with a value of No Data (All of our cells should contain data anyway!)

' Create a filename for the zone statistics
' We can look at the zone tables to find out the standard deviation values
' computed in each case!
```

```
zoneFN = av.GetProject.GetWorkDir.MakeTmp("zone","dbf")

' ********************************************

' Create the Zonal Statistics Table for surfDiff values
statsSurfVTab = (surfDiff).ZonalStatsTable(zoneGrid, Prj.MakeNull, zoneField, true, zoneFN)

' Check if problems creating the statsSurfVTab
if (statsSurfVTab.HasError) then
   MsgBox.Error("The statsSurfVTab has an error!","")
   exit
end

'
************************************************************************
**

' Find the sum of the squares of "denseSurf - sparseSurf" (or surfDiff) residuals
sumSurfDiffSqr = (surfDiffSqr).ZonalStats(#GRID_STATYPE_SUM, zoneGrid,
Prj.MakeNull,
                                  zoneField, true)

if (sumSurfDiffSqr.HasError) then
   MsgBox.Error("The sumSurfDiffSqr had an error!","")
   exit
end

' Retrieve the sum value stored in each cell, without loosing precision of the number
' (We will divide by 10000 later to account for this step)
multSumSurfDiffSqr = (sumSurfDiffSqr * 10000.AsGrid).Int

sumSurfVTab = multSumSurfDiffSqr.GetVTab
if (sumSurfVTab = nil) then
   MsgBox.Info("Could not get the sumSurfVTab!","")
   exit
end

' Identify the "value" field of the individual cells in the sumSurfVTab
sumFieldSurf = sumSurfVTab.FindField("Value")

' Retrieve the value which is in each of the cells of the sumSurfVTab
multSumSurfDiffSqrValue = sumSurfVTab.ReturnValue(sumFieldSurf, 0)
```

```
'  Divide multSumSurfDiffSqr by 10000, since earlier we multiplied
'  the number by this value in order to retain precision
sumSurfDiffSqrValue = multSumSurfDiffSqrValue/10000

'
***********************************************************************
**

'  Find the number of cells in the surfDiffSqr grid.
'  Find number of rows and number of columns and multiply
rcList = surfDiffSqr.GetNumRowsAndCols
numRows = rcList.Get(0)
'  MsgBox.Info("The number of rows is "+numRows.AsString,"Rows")
numCols = rcList.Get(1)
'  MsgBox.Info("The number of cols is "+numCols.AsString,"Cols")
numCells = numRows * numCols
'  MsgBox.Info("The number of cells is "+numCells.AsString,"Total number of Cells")

'
***********************************************************************

'  RMS (root mean square) is computed across all cells of the difference surface
RMSsurf = (sumSurfDiffSqrValue/numCells).Sqrt

return RMSsurf
******************************************

******************************************
'Name:  Proj.fast_tab_stats
'Author/Date:  Pam LaFontaine / 20 May 98
'Updated_by/Date:  11 Aug 99
'Course:          URP6272, Spring 1998
'Prepared at:      GeoPlan Center, University of Florida
'--------------------------------------------------
'Description:  Computes a RMS statistics to compare two point sets
'Requires:
'Calls:  Proj.make_fast_grid, Proj.make_fast_surf, and Proj.fast_tab_stats_calc
'Is called by:  Proj.comp_table
'Self:  theTheme = SELF.Get(0), fthm = SELF.Get(1)
'Returns:  RMSsurf
'FileName:  Proj.fast_tab_stats
'--------------------------------------------------
```

```
'  The original data set's theme is theTheme and the thinned data set's theme is fthm
theTheme = SELF.Get(0)
fthm = SELF.Get(1)

theFTab1 = theTheme.GetFTab
theFTab2 = fthm.GetFTab
FTab1count = theFTab1.GetNumRecords
FTab2count = theFTab2.GetNumRecords

'  The way this is set up, FTab1 will always be the "dense" data set!
if (FTab1count >= FTab2count) then

   denseSurfaceGridThm = av.Run("Proj.make_fast_surf", theTheme)
   denseSurfaceGrid = denseSurfaceGridThm.GetGrid

   sparseSurfaceGridThm = av.Run("Proj.make_fast_surf", fthm)
   sparseSurfaceGrid = sparseSurfaceGridThm.GetGrid
else  '  should never happen
   MsgBox.Info("This should never happen!","")
      exit
end

surfList = {denseSurfaceGrid, sparseSurfaceGrid}

RMSsurf = av.Run("Proj.fast_tab_stats_calc", surfList)

return RMSsurf
******************************************

******************************************
'Name:  Proj.make_surface
'Author/Date:  Pam LaFontaine / 25 April 98
'Updated_by/Date:  4 July 99
'Course:          URP6272, Spring 1998
'Prepared at:      NIMA Bethesda, MD
'-------------------------------------------------
'Description:  Creates a surface from a shapefile point set
'Requires:
'Calls:
'Is called by:  Proj.statistics
'Self:  theTheme   (a theme from Proj.statistics)
'Returns:  gthm
```

```
'FileName:  Proj.make_surface
'---------------------------------------------------

' Spatial.Surface
' This system script is used to create a surface for an active point or multipoint FTab
' in ArcView GIS Version 3.1
' It has been specially edited to make grids (interpolate surfaces) for the
' SHOALS Surface Simplification program.

theView = av.GetActiveDoc

'  Pass a theme from Proj.statistics
theTheme = SELF

' All spatial analysis takes place within an environment that is defined
' by an extent rectangle, a cell size and a raster mask. The default analysis
' environment corresponds to the maximum extent of the input Grids,
' the maximum (coarsest) cell size of the input Grids and no mask.

' obtain extent and cell size if not set
ae = theView.GetExtension(AnalysisEnvironment)
box = Rect.Make(0@0,1@1)
cellSize = 1
if ((ae.GetExtent(box) <> #ANALYSISENV_VALUE) or (ae.GetCellSize(cellSize) <>
#ANALYSISENV_VALUE)) then
  'ce = AnalysisPropertiesDialog.Show(theView,TRUE,"Output Grid Specification")
  'if (ce = NIL) then
  '  return NIL
  'end
  'ce.GetCellSize(cellSize)
  'ce.GetExtent(box)
end

' convert z or m shapes to non-z or non-m shapes,
' just create the empty shapefile with all fields
' here to populate interp dialog
t = theTheme
theFTab = theTheme.GetFTab
theClassName = theFTab.GetShapeClass.GetClassName
needDelete = FALSE
if ((theClassName = "PointZ") or
    (theClassName = "MultiPointZ") or
    (theClassName = "PointM") or
```

```
    (theClassName = "MultiPointM")) then
  needDelete = TRUE
  theFieldList = theFTab.GetFields.Clone
  theShapeField = theFTab.FindField("Shape")
  theNewShapefile = av.GetProject.GetWorkDir.MakeTmp("sface","shp")
  theNewFTab = FTab.MakeNew(theNewShapefile,Point)
  theFieldList.RemoveObj(theShapeField) 'ignore the shape field
  theFieldCount = theFieldList.Count - 1
  theNewShapeField = theNewFTab.FindField("Shape")
  aShape = theFTab.ReturnValue(theShapeField,0)
  hasZcoord = aShape.HasZ
  hasMcoord = aShape.IsMeasured
  if (hasZcoord) then
    zCoordField = Field.Make("ShapeZ",#FIELD_DOUBLE,12,3)
    theNewFTab.AddFields({zCoordField})
  end
  if (hasMcoord) then
    mCoordField = Field.Make("ShapeM",#FIELD_DOUBLE,12,3)
    theNewFTab.AddFields({mCoordField})
  end
  theNewFieldList = theFieldList.DeepClone
  theNewFTab.AddFields(theNewFieldList)
  t = FTheme.Make(theNewFTab)
end

'  The below code was modified so that a dialog would not be required
'
**************************************************************************

' Get the depth field
zField = theFTab.FindField("Z")

'  Make IDW...setting aPower, aRadius, aBarrierFTab
'  Power of 2 used to determine influence of surrounding points, no radius (default of
'  12 points used with no max distance), no barrier set.
anInterp = Interp.MakeIDW(2, nil, nil)
'
**************************************************************************

' export z and m shapefile to non-z or non-m shapefile
' export only the selected set
if ((theClassName = "PointZ") or
    (theClassName = "MultiPointZ") or
```

```
  (theClassName = "PointM") or
  (theClassName = "MultiPointM")) then
av.ClearMsg
av.ClearStatus
av.ShowStopButton
av.ShowMsg("Exporting Shapes...")
theIndex = 0
theBitMap = theFTab.GetSelection
if (theBitMap.Count = 0) then
  numFeatures = theFTab.GetNumRecords
  theBitMap.SetAll
  unsetBitmap = TRUE 'reset flag for end of loop
else
  numFeatures = theFTab.GetNumSelRecords
  unsetBitmap = FALSE
end
done = FALSE
offset = -1
while (not done)
  rec = theBitmap.GetNextSet(offset)
  offset = rec
  if (rec <> -1) then
    theShape = theFTab.ReturnValue(theShapeField,rec)
    if ((theClassName = "PointZ") or
        (theClassName = "PointM")) then
      if (zField.GetName = "ShapeZ") then
        theZ = theShape.GetZ
        if (theZ.IsNull.Not) then
          theNewRecnum = theNewFTab.AddRecord
          theNewFTab.SetValue(zCoordField,theNewRecnum,theZ)
          theShape = theShape.AsPoint
          theNewFTab.SetValue(theNewShapeField,theNewRecnum,theShape)
        end
      elseif (zField.GetName = "ShapeM") then
        theM = theShape.GetM
        if (theM.IsNull.Not) then
          theNewRecnum = theNewFTab.AddRecord
          theNewFTab.SetValue(mCoordField,theNewRecnum,theM)
          theShape = theShape.AsPoint
          theNewFTab.SetValue(theNewShapeField,theNewRecnum,theShape)
        end
      else
        theValue = theFTab.ReturnValue(theFTab.FindField(zField.GetName),rec)
```

```
    if (theValue.IsNull.Not) then
      theNewRecnum = theNewFTab.AddRecord
      theNewFTab.SetValue(zField,theNewRecnum,theValue)
      theShape = theShape.AsPoint
      theNewFTab.SetValue(theNewShapeField,theNewRecnum,theShape)
    end
  end
elseif ((theClassName = "MultiPointZ") or
        (theClassName = "MultiPointM")) then
  for each p in theShape.AsList
    if (zField.GetName = "ShapeZ") then
      theZ = p.GetZ
      if (theZ.IsNull.Not) then
        theNewRecnum = theNewFTab.AddRecord
        theNewFTab.SetValue(zCoordField,theNewRecnum,theZ)
        p = p.AsPoint
        theNewFTab.SetValue(theNewShapeField,theNewRecnum,p)
      end
    elseif (zField.GetName = "ShapeM") then
      theM = p.GetM
      if (theM.IsNull.Not) then
        theNewRecnum = theNewFTab.AddRecord
        theNewFTab.SetValue(mCoordField,theNewRecnum,theM)
        p = p.AsPoint
        theNewFTab.SetValue(theNewShapeField,theNewRecnum,p)
      end
    else
      theValue = theFTab.ReturnValue(theFTab.FindField(zField.GetName),rec)
      if (theValue.IsNull.Not) then
        theNewRecnum = theNewFTab.AddRecord
        theNewFTab.SetValue(zField,theNewRecnum,theValue)
        p = p.AsPoint
        theNewFTab.SetValue(theNewShapeField,theNewRecnum,p)
      end
    end
  end
end
theIndex = theIndex + 1
progress = (theIndex/numFeatures) * 100
doMore = av.SetStatus(progress)
if (not doMore) then
  theNewFTab.SetEditable(FALSE)
  if (unsetBitmap) then
```

```
        theBitmap.ClearAll
      end
      t = NIL
      theFTab.DeActivate
      theFTab = NIL
      theNewFTab.DeActivate
      theNewFTab = NIL
      av.PurgeObjects
      File.Delete(theNewShapefile)
      theNewShapefile.SetExtension("dbf")
      File.Delete(theNewShapefile)
      theNewShapefile.SetExtension("shx")
      File.Delete(theNewShapefile)
      return NIL
    end
  else
    done = TRUE
  end
 end
 theNewFTab.Flush
 if (unsetBitmap) then
  theBitmap.ClearAll
 end
 if (theNewFTab.GetNumRecords = 0) then
  MsgBox.Error(zField.GetName++" is null for all features","Interpolate Grid")
  theNewFTab.SetEditable(FALSE)
  t = NIL
  theFTab.DeActivate
  theFTab = NIL
  theNewFTab.DeActivate
  theNewFTab = NIL
  av.PurgeObjects
  File.Delete(theNewShapefile)
  theNewShapefile.SetExtension("dbf")
  File.Delete(theNewShapefile)
  theNewShapefile.SetExtension("shx")
  File.Delete(theNewShapefile)
  return NIL
 end
 theFTab = theNewFTab
end

' perform interpolation
```

```
aPrj = theView.GetProjection
r = Grid.MakeByInterpolation(theFTab,aPrj,zField,anInterp,{cellSize, box})

' delete temp shapefile if created
if (needDelete) then
  theNewFTab.SetEditable(FALSE)
  t = NIL
  theFTab.DeActivate
  theFTab = NIL
  theNewFTab.DeActivate
  theNewFTab = NIL
  av.PurgeObjects
  File.Delete(theNewShapefile)
  theNewShapefile.SetExtension("dbf")
  File.Delete(theNewShapefile)
  theNewShapefile.SetExtension("shx")
  File.Delete(theNewShapefile)
end

' rename data set
aFN = av.GetProject.GetWorkDir.MakeTmp("sface","")
r.Rename(aFN)

' check if output is ok
if (r.HasError) then
  return NIL
end

' create a theme
gthm = GTheme.Make(r)

' create appropriate legend for theme
theLegend = gthm.GetLegend
theLegend.Interval(gthm,"Value",9)
theSymbolList = theLegend.GetSymbols
theNullSymbol = theSymbolList.Get(theSymbolList.Count - 1)
theSymbolList.Remove(theSymbolList.Count - 1)
startColor = Color.Make
startColor.SetRgbList({90,90,90})
theSymbolList.RampColors(startColor,Color.GetWhite)
theSymbolList.Add(theNullSymbol)
gthm.UpdateLegend
```

```
' set name for theme
gthm.SetName("Surface from " + theTheme.GetName)

' add theme to the specifiedView
'theView.AddTheme(gthm)

' set self for surface if done from scene
if (theView.GetClass.GetClassName = "Scene") then
  av.Run("3D.SetSurface",{gthm})
end

return gthm
*******************************************

*******************************************
'Name:  Proj.make_theme
'Author/Date:  Pam LaFontaine / 20 May 98
'Updated_by/Date:  22 Jan 99
'Course:          URP6272, Spring 1998
'Prepared at:      GeoPlan Center, University of Florida
'-------------------------------------------------
'Description:  Creates a shapefile from theBitmap (selected new data set values),
'              but does not add the new theme to the view.
'Requires:  One active shapefile theme
'Calls:
'Is called by:  Proj.fast_make_area_1s
'Self:  theView = SELF.Get(0), t = SELF.Get(1)
'Returns:
'FileName:  Proj.make_theme
'-------------------------------------------------

'  Get the active view and the active theme
theView = SELF.Get(0)
t = SELF.Get(1)
' ****************************************

if (t.Is( FTHEME ).Not) then
   if (t.CanExportToFtab.Not) then
      return NIL
   end
end

if (t.Is (FTHEME).Not) then
```

```
    def = av.GetProject.MakeFileName("theme", "shp")
    'def = FileDialog.Put(def, "*.shp", "Convert " + t.getName)
    if (def = NIL) then
      return NIL
    end
    anFTab = t.ExportToFtab(def)
  else
    tbl = t.GetFTab
    attribVis = FALSE
    for each f in tbl.GetFields
      if ((f.IsVisible) and not (f.IsTypeShape)) then
        attribVis = TRUE
        break
      end
    end

    shapeVis = tbl.FindField("Shape").IsVisible
    if ((attribVis and shapeVis).Not) then
      return NIL
    end

    def = av.GetProject.MakeFileName("theme", "shp")
    'def = FileDialog.Put(def, "*.shp", "Convert " + t.getName)
    if (def = NIL) then
      return NIL
    end

    shpfld = (tbl.FindField("Shape"))
    if (shpfld.IsVisible.Not) then
      shpfld.SetVisible(shpfld.IsVisible.Not)
      WasNotVisible = TRUE
    else
      WasNotVisible =FALSE
    end

    anFTab = tbl.Export(def, Shape, tbl.GetSelection.Count > 0)
    if (anFTab.HasError) then
      if (anFTab.HasLockError) then
        MsgBox.Error("Unable to acquire Write Lock for file " + def.GetBaseName,"")
      else
        MsgBox.Error("Unable to create " + def.GetBaseName,"")
      end
      return NIL
```

```
    end

    if (WasNotVisible) then
        shpfld.SetVisible(FALSE)
    end
end

'create a theme
fthm = FTheme.Make(anFTab)
'theView.AddTheme(fthm)
'bring the View to the front
'theView.GetWin.Activate

'  Return fthm to Proj.make_fast_area_1s to be used
'  without adding the new theme to the view.
return fthm
*******************************************

*******************************************
'Name:  Proj.make_area_1s
'Author/Date:  Pam LaFontaine / 22 April 98
'Updated_by/Date:  16 Jan 99
'Course:          URP6272, Spring 1998
'Prepared at:      GeoPlan Center, University of Florida
'--------------------------------------------------
'Description:  Changes the "area_pts" value for user's selected points to 1 and then
'              runs the Proj.thin subroutine and the Proj.add_theme subroutine
'Requires:  One active shapefile theme
'Calls:  Proj.thin and Proj.add_theme
'Is called by:  (Menu request -- Run Thin)
'Self:
'Returns:
'FileName:  Proj.make_area_1s
'--------------------------------------------------

'  Get the active view and the active theme
theView = av.GetActiveDoc
theTheme = theView.GetActiveThemes.Get(0)
theVTab = theTheme.GetFTab

theViewName = theView.GetName
theThemeName = theTheme.GetName
```

```
'  Add the selected portion of the data set to the view as a theme
if (_portion) then
   MsgBox.Info("Since you've selected only a portion of the original data set, that portion"++
            "will be added to the view as a theme.","")
   av.Run("Proj.add_theme", {theView, theTheme})
   _portion = false
end

if ((theTheme = nil) OR (theView = nil) OR (theVTab = nil)) then
   MsgBox.Error("You must have one view and one theme active to run this function!","")
   exit
else
   ans = MsgBox.YesNo("Are you thinning a selected area of the  "+theThemeName.AsString+
   " theme of the "+theView.AsString+" view?","Are you sure?", true)
   if (ans = false) then
      exit
   end
end

'  Find the "area_pts" fields in the table
_areaField = theVTab.FindField("area_pts")

theBitmap = theVTab.GetSelection

'  Give a value of "1" to the "area_pts" field of records in the selected area
if (theVTab.StartEditingWithRecovery) then
   for each rec in theBitmap
      theVTab.BeginTransaction
      theVTab.SetValue(_areaField, rec, 1)
      theVTab.EndTransaction
   end
else
  MsgBox.Warning("Problem Editing Table","")
  exit
end

MsgBox.Info("The number of selected points is "+theBitmap.Count.AsString,"")
_oldRecValue = theBitmap.Count

theVTab = av.Run("Proj.thin", {theVTab, theTheme})

theBitmap = theVTab.GetSelection
theBitmap.Clearall
```

```
theVTab.UpdateSelection

'  Use queries to select only the new set of points
theQuery1 = "([area_pts] = 1)"
theVTab.Query(theQuery1, theBitmap, #VTAB_SELTYPE_NEW)
theVTab.UpdateSelection

theQuery2 = "([mark] = 0)"
theVTab.Query(theQuery2, theBitmap, #VTAB_SELTYPE_AND)
theVTab.UpdateSelection

MsgBox.Info("The number of points in the new set is "+theBitmap.Count.AsString,"")

'  Allow the user to create a shapefile from the new data set
av.Run("Proj.add_theme", {theView, theTheme})

theBitmap = theVTab.GetSelection
theBitmap.Clearall
theVTab.UpdateSelection

theVTab.StopEditingWithRecovery(false)
*******************************************

*******************************************
'Name:  Proj.make_fast_surf
'Author/Date:  Pam LaFontaine / 16 May 98
'Updated_by/Date:  11 Aug 99
'Course:          URP6272, Spring 1998
'Prepared at:     GeoPlan Center, University of Florida
'---------------------------------------------------
'Description:  Creates a surfaces from a shapefile point set using a specified interpolation
' method  (currently set to IDW vice Spline)
'Requires:
'Calls:
'Is called by:  Proj.fast_stats or Proj.fast_tab_stats
'Self:  t   (a theme from Proj.fast_stats or Proj.fast_tab_stats)
'Returns:  gthm
'FileName:  Proj.make_fast_surf
'---------------------------------------------------

' Spatial.Surface
' This system script is used to create a surface for an active point or multipoint FTab
' in ArcView GIS Version 3.1
```

```
' It has been specially edited to make grids (interpolate surfaces) for the
' SHOALS Surface Simplification program.

theView = av.GetActiveDoc

'  Pass a theme from Proj.statistics
theTheme = SELF

' All spatial analysis takes place within an environment that is defined
' by an extent rectangle, a cell size and a raster mask. The default analysis
' environment corresponds to the maximum extent of the input Grids,
' the maximum (coarsest) cell size of the input Grids and no mask.

' obtain extent and cell size if not set
ae = theView.GetExtension(AnalysisEnvironment)
box = Rect.Make(0@0,1@1)
cellSize = 1
if ((ae.GetExtent(box) <> #ANALYSISENV_VALUE) or (ae.GetCellSize(cellSize) <>
#ANALYSISENV_VALUE)) then
  'ce = AnalysisPropertiesDialog.Show(theView,TRUE,"Output Grid Specification")
  'if (ce = NIL) then
  '  return NIL
  'end
  'ce.GetCellSize(cellSize)
  'ce.GetExtent(box)
end

' convert z or m shapes to non-z or non-m shapes,
' just create the empty shapefile with all fields
' here to populate interp dialog
t = theTheme
theFTab = theTheme.GetFTab
theClassName = theFTab.GetShapeClass.GetClassName
needDelete = FALSE
if ((theClassName = "PointZ") or
    (theClassName = "MultiPointZ") or
    (theClassName = "PointM") or
    (theClassName = "MultiPointM")) then
  needDelete = TRUE
  theFieldList = theFTab.GetFields.Clone
  theShapeField = theFTab.FindField("Shape")
  theNewShapefile = av.GetProject.GetWorkDir.MakeTmp("sface","shp")
  theNewFTab = FTab.MakeNew(theNewShapefile,Point)
```

```
  theFieldList.RemoveObj(theShapeField) 'ignore the shape field
  theFieldCount = theFieldList.Count - 1
  theNewShapeField = theNewFTab.FindField("Shape")
  aShape = theFTab.ReturnValue(theShapeField,0)
  hasZcoord = aShape.HasZ
  hasMcoord = aShape.IsMeasured
  if (hasZcoord) then
    zCoordField = Field.Make("ShapeZ",#FIELD_DOUBLE,12,3)
    theNewFTab.AddFields({zCoordField})
  end
  if (hasMcoord) then
    mCoordField = Field.Make("ShapeM",#FIELD_DOUBLE,12,3)
    theNewFTab.AddFields({mCoordField})
  end
  theNewFieldList = theFieldList.DeepClone
  theNewFTab.AddFields(theNewFieldList)
  t = FTheme.Make(theNewFTab)
end

'  The below code was modified so that a dialog would not be required
'
**************************************************************************
' Get the depth field
zField = theFTab.FindField("Z")

'  Make IDW...setting aPower, aRadius, aBarrierFTab
'  Power of 2 used to determine influence of surrounding points, no radius (default of
'  12 points used with no max distance), no barrier set.
anInterp = Interp.MakeIDW(2, nil, nil)
'
**************************************************************************
' export z and m shapefile to non-z or non-m shapefile
' export only the selected set
if ((theClassName = "PointZ") or
    (theClassName = "MultiPointZ") or
    (theClassName = "PointM") or
    (theClassName = "MultiPointM")) then
  av.ClearMsg
  av.ClearStatus
  av.ShowStopButton
  av.ShowMsg("Exporting Shapes...")
  theIndex = 0
  theBitMap = theFTab.GetSelection
```

```
if (theBitMap.Count = 0) then
  numFeatures = theFTab.GetNumRecords
  theBitMap.SetAll
  unsetBitmap = TRUE 'reset flag for end of loop
else
  numFeatures = theFTab.GetNumSelRecords
  unsetBitmap = FALSE
end
done = FALSE
offset = -1
while (not done)
  rec = theBitmap.GetNextSet(offset)
  offset = rec
  if (rec <> -1) then
    theShape = theFTab.ReturnValue(theShapeField,rec)
    if ((theClassName = "PointZ") or
        (theClassName = "PointM")) then
      if (zField.GetName = "ShapeZ") then
        theZ = theShape.GetZ
        if (theZ.IsNull.Not) then
          theNewRecnum = theNewFTab.AddRecord
          theNewFTab.SetValue(zCoordField,theNewRecnum,theZ)
          theShape = theShape.AsPoint
          theNewFTab.SetValue(theNewShapeField,theNewRecnum,theShape)
        end
      elseif (zField.GetName = "ShapeM") then
        theM = theShape.GetM
        if (theM.IsNull.Not) then
          theNewRecnum = theNewFTab.AddRecord
          theNewFTab.SetValue(mCoordField,theNewRecnum,theM)
          theShape = theShape.AsPoint
          theNewFTab.SetValue(theNewShapeField,theNewRecnum,theShape)
        end
      else
        theValue = theFTab.ReturnValue(theFTab.FindField(zField.GetName),rec)
        if (theValue.IsNull.Not) then
          theNewRecnum = theNewFTab.AddRecord
          theNewFTab.SetValue(zField,theNewRecnum,theValue)
          theShape = theShape.AsPoint
          theNewFTab.SetValue(theNewShapeField,theNewRecnum,theShape)
        end
      end
    elseif ((theClassName = "MultiPointZ") or
```

```
      (theClassName = "MultiPointM")) then
  for each p in theShape.AsList
    if (zField.GetName = "ShapeZ") then
      theZ = p.GetZ
      if (theZ.IsNull.Not) then
        theNewRecnum = theNewFTab.AddRecord
        theNewFTab.SetValue(zCoordField,theNewRecnum,theZ)
        p = p.AsPoint
        theNewFTab.SetValue(theNewShapeField,theNewRecnum,p)
      end
    elseif (zField.GetName = "ShapeM") then
      theM = p.GetM
      if (theM.IsNull.Not) then
        theNewRecnum = theNewFTab.AddRecord
        theNewFTab.SetValue(mCoordField,theNewRecnum,theM)
        p = p.AsPoint
        theNewFTab.SetValue(theNewShapeField,theNewRecnum,p)
      end
    else
      theValue = theFTab.ReturnValue(theFTab.FindField(zField.GetName),rec)
      if (theValue.IsNull.Not) then
        theNewRecnum = theNewFTab.AddRecord
        theNewFTab.SetValue(zField,theNewRecnum,theValue)
        p = p.AsPoint
        theNewFTab.SetValue(theNewShapeField,theNewRecnum,p)
      end
    end
  end
end
theIndex = theIndex + 1
progress = (theIndex/numFeatures) * 100
doMore = av.SetStatus(progress)
if (not doMore) then
  theNewFTab.SetEditable(FALSE)
  if (unsetBitmap) then
    theBitmap.ClearAll
  end
  t = NIL
  theFTab.DeActivate
  theFTab = NIL
  theNewFTab.DeActivate
  theNewFTab = NIL
  av.PurgeObjects
```

```
      File.Delete(theNewShapefile)
      theNewShapefile.SetExtension("dbf")
      File.Delete(theNewShapefile)
      theNewShapefile.SetExtension("shx")
      File.Delete(theNewShapefile)
      return NIL
    end
  else
    done = TRUE
  end
 end
 theNewFTab.Flush
 if (unsetBitmap) then
  theBitmap.ClearAll
 end
 if (theNewFTab.GetNumRecords = 0) then
  MsgBox.Error(zField.GetName++" is null for all features","Interpolate Grid")
  theNewFTab.SetEditable(FALSE)
  t = NIL
  theFTab.DeActivate
  theFTab = NIL
  theNewFTab.DeActivate
  theNewFTab = NIL
  av.PurgeObjects
  File.Delete(theNewShapefile)
  theNewShapefile.SetExtension("dbf")
  File.Delete(theNewShapefile)
  theNewShapefile.SetExtension("shx")
  File.Delete(theNewShapefile)
  return NIL
 end
 theFTab = theNewFTab
end

' perform interpolation
aPrj = theView.GetProjection
r = Grid.MakeByInterpolation(theFTab,aPrj,zField,anInterp,{cellSize, box})

' delete temp shapefile if created
if (needDelete) then
 theNewFTab.SetEditable(FALSE)
 t = NIL
 theFTab.DeActivate
```

```
  theFTab = NIL
  theNewFTab.DeActivate
  theNewFTab = NIL
  av.PurgeObjects
  File.Delete(theNewShapefile)
  theNewShapefile.SetExtension("dbf")
  File.Delete(theNewShapefile)
  theNewShapefile.SetExtension("shx")
  File.Delete(theNewShapefile)
end

' rename data set
aFN = av.GetProject.GetWorkDir.MakeTmp("sface","")
r.Rename(aFN)

' check if output is ok
if (r.HasError) then
  return NIL
end

' create a theme
gthm = GTheme.Make(r)

' create appropriate legend for theme
theLegend = gthm.GetLegend
theLegend.Interval(gthm,"Value",9)
theSymbolList = theLegend.GetSymbols
theNullSymbol = theSymbolList.Get(theSymbolList.Count - 1)
theSymbolList.Remove(theSymbolList.Count - 1)
startColor = Color.Make
startColor.SetRgbList({90,90,90})
theSymbolList.RampColors(startColor,Color.GetWhite)
theSymbolList.Add(theNullSymbol)
gthm.UpdateLegend

' set name for theme
gthm.SetName("Surface from " + theTheme.GetName)

' add theme to the specifiedView
'theView.AddTheme(gthm)

' set self for surface if done from scene
if (theView.GetClass.GetClassName = "Scene") then
```

```
  av.Run("3D.SetSurface",{gthm})
end

return gthm
******************************************

******************************************
'Name:  Proj.random_thin
'Author/Date:  Pam LaFontaine / 03 Jun 99
'Updated_by/Date:  19 Aug 99
'Course:          Spring 1999
'Prepared at:      NIMA
'---------------------------------------------------
'Description:  Gets the VTab associated with the view's active theme.
'              Selects records in a given fashion (in accordance with option chosen).
'              Makes a new shapefile of all selected records.
'              This represents the new systematic or random data set.
'Requires:
'Calls:  Proj.add_theme
'Is called by:
'Self:
'Returns:
'FileName:  Proj.random_thin
'---------------------------------------------------

'  Get the active view and the active theme
theView = av.GetActiveDoc
theTheme = theView.GetActiveThemes.Get(0)
theVTab = theTheme.GetFtab

theViewName = theView.GetName
theThemeName = theTheme.GetName

'  Select all of the points in the selected theme.
theBitmap = theVTab.GetSelection
theBitmap.SetAll
theVTab.UpdateSelection

MsgBox.Info("The number of points in this theme is "+theBitmap.Count.AsString,"")
orig = theBitmap.Count

'  Find the "mark", "area_pts" and "Z" fields in the table
_markField = theVTab.FindField("mark")
```

```
_areaField = theVTab.FindField("area_pts")
_zField = theVTab.FindField("Z")

MsgBox.Report("The following process will allow you to select a subset of points from your"++
            "selected theme.  In this manner you may, either systematically, or via a"++
            "random process, create a subset of your original data which consists of"++
            "the same number of points as the thinned data set generated by this program's"++
            "thinning algorithm.  The accuracy of the two equally-sized 'thinned' data sets"++
            "can then be compared using the 'Run Statistics' procedure.",
            "About Systematic and Random Thinning")

'  Allow user to select the type of input ... either a percentage to be removed or a number
'  of points to be removed.
optionList = {"% of data to be removed?", "# of points to be removed?"}

option = MsgBox.ChoiceAsString(optionList, "Would you like to enter ...", "Pick An Option")
if (option = nil) then
   exit
elseif (option = optionList.Get(0)) then
   pcent = MsgBox.Input("What is the percentage of data that you wish to remove from this
set?",
                  "Enter a number value for the percentage","")
   if (pcent.IsNumber.Not) then
      MsgBox.Error("Your percentage value must be a number!  No symbols or letters are
allowed!","")
      exit
   end
   '  The value in the variable "pcent" equals the percentage of data to be removed
   ans = MsgBox.YesNo("You have chosen to remove"++pcent.AsString+"% of the original
points.",
                   "Is this correct?", true)
   if (ans = false) then
     exit
   end
   '  the variable "remove" will equal the number of points (or records) to be removed from the
   '  original data set.
   remove = (pcent.AsNumber / 100) * orig
else
   points = MsgBox.Input("How many points do you wish to remove from this data set?",
                  "Enter a number value for the points to be removed","")
   if (points.IsNumber.Not) then
      MsgBox.Error("Your point value must be a number!  No symbols or letters are
allowed!","")
```

```
      exit
   end
   ' The value in the variable "points" equals the number of points to be removed
   ans = MsgBox.YesNo("You have chosen to remove"++points.AsString++"of the original
points.",
                  "Is this correct?", true)
   if (ans = false) then
     exit
   end
   ' This means that the variable "points" equals the number of data points to be removed
   remove = points.AsNumber
end

if (remove >= orig) then
   MsgBox.Info("You can not remove all of the points from your original data set!","ERROR")
   exit
end

' **************************************************************
' The variable "remain" holds the number of points (or records) which will be in the subset
remain = orig - remove
' **************************************************************

optionList2 = {"...use systematic method?", "...use random method?"}

option2 = MsgBox.ChoiceAsString(optionList2, "Approximately"++remain.AsString++"points
will be left after"++
        remove.AsString++"points are removed.  Would you like to...", "Pick An Option")
if (option2 = nil) then
   exit
elseif (option2 = optionList2.Get(0)) then
   ' A systematic method has been chosen
   theCount = 0
   countnum = 0
   theStep = (orig / remove)
   theBitmap = theVTab.GetSelection
   theBitmap.Clearall
   theVTab.UpdateSelection
   for each index in theVTab
      ' MsgBox.Info("Index value ="++index.AsString,"Index Value")
      if (index = theCount.Round) then
         ' Put the index point itself into the bitmap set
         theBitmap.Set(index)
```

```
            theVTab.UpdateSelection
            theCount = theCount + theStep
            countnum = countnum + 1
        end
        if (countnum <> remove) then
            continue
        else
            break
        end
    end
    '  MsgBox.Info("You made it this far!","WOW")
    'theTable = av.GetActiveDoc
    'theTable.GetVTab.GetSelection.Not
    theVTab.GetSelection.Not
    'theTable.GetVTab.UpdateSelection
    theVTab.UpdateSelection
    av.Run("Proj.add_theme", {theView, theTheme})
    theBitmap = theVTab.GetSelection
    theBitmap.Clearall
    theVTab.UpdateSelection
else
    '  A random method has been chosen
    theBitmap = theVTab.GetSelection
    theBitmap.Clearall
    theVTab.UpdateSelection
    '  Create a list consisting of sequential numbers from 1 to the number of points in
    '  the original set of data (the variable "orig" is defined above).  This list will
    '  be randomly scrambled or "shuffled" like a deck of cards.
    theList = list.make
    total = orig-1
    for each num in 0..total
        theList.Add(num+1)
    end
    numberOfObjects = theList.Count
    '  If the number of objects in your list is not equal to the value "orig",
    '  then you have a problem
    if (numberOfObjects <> orig) then
        MsgBox.Error("Your list should have"++orig.AsString++"objects."+NL+
                "Only"++numberOfObjects.AsString++"objects were stored!","BAD LIST!")
        exit
    end
    if (theList.Get(0) <> 1) then
        MsgBox.Error("Your list is messed up!","BAD LIST!")
```

```
    exit
end
' Shuffle numbers until satisfied...
keepgoing = TRUE
while (keepgoing)
    for each card in 0..total
        randnum = Number.MakeRandom(0,total)
        randRecordValue = theList.Get(randnum)
        swapRecordValue = theList.Get(card)
        theList.Set(card, randRecordValue)
        theList.Set(randnum, swapRecordValue)
    end
    ans = MsgBox.YesNo("Do you wish to stop the 'shuffling' process?",
                    "Stop Shuffling?", true)
    if (ans) then
        ' Allow for breaking out of while loop
        keepgoing = false
    end
end
' Now make a list of those random records you wish to keep out of the total
' original set. You should have "remain" (variable defined above) random records
' in this dataset.
' Create a counter
theCount = 0
remainExtra = remain+10
for each i in 0..remainExtra
    loopnum = i+1
    pickRec = theList.Get(i)
    for each index in theVTab
        if (index = pickRec) then
            theBitmap.Set(index)
            theVTab.UpdateSelection
            theCount = theCount + 1
            break
        end
    end
    'if (i+1<>theCount) then
    '    ans = MsgBox.YesNo("A random index value was not found."+NL+
    '                    "Do you wish to abort the loop?", "ABORT LOOP?", true)
    '    if (ans) then
    '        exit
    '    end
    'end
```

```
      ' Emergency exit
      if (i = (remain/2).truncate) then
         ans = MsgBox.YesNo("Half-Way Through...Do you wish to abort the loop?",
"ABORT LOOP?", false)
         if (ans) then
            exit
         end
      end
      if (theCount = remain) then
         break
      end
   end
   MsgBox.Info("The for-loop had to run through"++loopnum.AsString++"iterations!","Number
of Iterations")
   theVTab.GetSelection
   theVTab.UpdateSelection
   av.Run("Proj.add_theme", {theView, theTheme})
   theBitmap = theVTab.GetSelection
   theBitmap.Clearall
   theVTab.UpdateSelection
end
 *******************************************

*******************************************
'Name:  Proj.run_many_thins
'Author/Date:  Pam LaFontaine / 16 May 98
'Updated_by/Date:  22 Jan 98
'Course:          URP6272, Spring 1998
'Prepared at:      GeoPlan Center, University of Florida
'--------------------------------------------------
'Description:  Computes a table of statistics to compare the surfaces fit to two point sets
'Requires:
'Calls:  Proj.fast_make_area_1s and Proj.fast_stats (and Proj.add_theme if required)
'Is called by:  (Menu request -- Run Multiple Thins)
'Self:
'Returns:
'FileName:  Proj.run_many_thins
'--------------------------------------------------

'  Set flag to direct program to proper subroutines (Will add the new data sets to the view for
'  comparison with "Comparison Table")
_compFlag = false
```

```
'  Get the active view and the active theme
theView = av.GetActiveDoc
theTheme = theView.GetActiveThemes.Get(0)
MsgBox.Warning("This function is designed to work on the theme listed at the top of the
view!","")

'  Allows you to go back out to the menu window to set the Analysis Properties to
'  match the Area of Interest!  If this is not done, your process will BOMB!
ans = MsgBox.YesNo("Have you set your Analysis Properties to correspond with the AOI?  ",
     "First Set Analysis Properties", true)
if (ans = false) then
   exit
end

'  Make a dictionary to hold all of the radius and tolerance pairs for each run
dataDict = Dictionary.Make(11)

'  Add the selected portion of the data set to the view as a theme
if (_portion) then
   MsgBox.Info("Since you've selected only a portion of the original data set, that portion"++
            "will be added to the view as a theme.","")
   av.Run("Proj.add_theme", {theView, theTheme})
   _portion = false
end

'  Select the working theme and the working VTab for these runs
theTheme = theView.GetActiveThemes.Get(0)
theVTab = theTheme.GetFTab

'  Find the "area_pts" fields in the table
_areaField = theVTab.FindField("area_pts")

theBitmap = theVTab.GetSelection

theSelection = theBitmap.count
'  If user has failed to select an area first, exit!
if ((theSelection = NIL) OR (theSelection <= 0)) then
   MsgBox.Info("Must select an area to be thinned first...use Select Area option.", "")
   exit
end

'  Give a value of "1" to "area_pts" field of records in the selected area
if (theVTab.StartEditingWithRecovery) then
```

```
    for each rec in theBitmap
        theVTab.BeginTransaction
        theVTab.SetValue(_areaField, rec, 1)
        theVTab.EndTransaction
    end
else
   MsgBox.Warning("Problem Editing Table","")
   exit
end

'MsgBox.Info("The number of selected points is "+theBitmap.Count.AsString,"")
originalRecValue = theBitmap.Count

numRuns = MsgBox.Input("How many times do you want to run 'Thin' and 'Stats'?",
                        "Number of Iterations", "")

if ((numRuns = nil) or (numRuns <=0)) then
    exit
end

labelsList = {"radius (in feet):  ", "tolerance (in feet):  "}
defaultList = {"0","0"}
dataDict.Empty

for each run in 1..numRuns.AsNumber
    dataList = MsgBox.MultiInput("Enter the radius and tolerance for run #"++run.AsString,
"Enter Run Data", labelsList, defaultList)

    if ((dataList.Get(0) = nil) or (dataList.Get(0) < 0)) then
        exit
    elseif ((dataList.Get(1) = nil) or (dataList.Get(1) < 0)) then
        exit
    end

    dataDict.Add(run.AsString, dataList)
end

'  Get a file name for your new table
aFileName = FileDialog.Put("Comparison_Table.dbf".AsFileName, "*.dbf", "Create the
Comparison Table")
if (nil = aFileName) then
    exit
end
```

```
'  Create A dBase table to hold results of comparison runs
finalVTab = VTab.MakeNew (aFileName, dBase)

finalVTab.SetEditable(true)
runField = Field.Make("Run No", #FIELD_LONG, 6, 0)
radField = Field.Make("radius", #FIELD_FLOAT, 8, 3)
tolField = Field.Make("tolerance", #FIELD_FLOAT, 8, 3)
denseField = Field.Make("dense", #FIELD_LONG, 10, 0)
thinField = Field.Make("thin", #FIELD_LONG, 10, 0)
RMSsurfField = Field.Make("surf RMS", #FIELD_FLOAT, 10, 6)
MinDiffField = Field.Make("Min Diff", #FIELD_FLOAT, 8, 3)
MaxDiffField = Field.Make("Max Diff", #FIELD_FLOAT, 8, 3)
StdDevField = Field.Make ("Std Dev", #FIELD_FLOAT, 10, 6)
finalVTab.AddFields({runField, radField, tolField, denseField, thinField,
                RMSsurfField, MinDiffField, MaxDiffField, StdDevField})

for each run in 1..numRuns.AsNumber
   runValue = run
   radValue = dataDict.Get(run.AsString).Get(0).AsNumber
   tolValue = dataDict.Get(run.AsString).Get(1).AsNumber
   fthm = av.Run("Proj.fast_make_area_1s", {theView, theTheme, theVTab, radValue,
tolValue})
   _denseValue = originalRecValue
   _thinValue = _newRecValue
   valueList = av.Run("Proj.fast_stats", run)
   minimum = valueList.Get(0)
   maximum = valueList.Get(1)
   sd = valueList.Get(2)
   RMSsurf = valueList.Get(3)

   rec = finalVTab.AddRecord
   finalVTab.SetValue(runField, rec, runValue)
   finalVTab.SetValue(radField, rec, radValue)
   finalVTab.SetValue(tolField, rec, tolValue)
   finalVTab.SetValue(denseField, rec, _denseValue)
   finalVTab.SetValue(thinField, rec, _thinValue)
   finalVTab.SetValue(RMSsurfField, rec, RMSsurf)
   finalVTab.SetValue(MinDiffField, rec, minimum)
   finalVTab.SetValue(MaxDiffField, rec, maximum)
   finalVTab.SetValue(StdDevField, rec, sd)
end
finalVTab.SetEditable(false)
```

```
theVTab.StopEditingWithRecovery(false)
*****************************************

*****************************************
'Name:  Proj.select
'Author/Date:  Pam LaFontaine / 22 April 98
'Updated_by/Date:  16 Jan 99
'Course:         URP6272, Spring 1998
'Prepared at:     GeoPlan Center, University of Florida
'--------------------------------------------------
'Description:  Allows the user to designate an area (within the view) on
'             which the program will conduct analysis.  The user's mouse
'             becomes a tool with which he can click and drag over the
'             points he wishes to select.  (The user also has the option
'             to select the entire data set.)
'Requires:  One active shapefile theme in a view
'Calls:
'Is called by:  (Menu request -- Select Area)
'Self:
'Returns:
'FileName:  Proj.select
'--------------------------------------------------

'  Get the active view and the active theme
theView = av.GetActiveDoc
theTheme = theView.GetActiveThemes.Get(0)
theVTab = theTheme.GetFtab

theViewName = theView.GetName
theThemeName = theTheme.GetName

theBitmap = theVTab.GetSelection
theBitmap.Clearall
theVTab.UpdateSelection

if ((theTheme = nil) OR (theView = nil) OR (theVTab = nil)) then
   MsgBox.Error("You must have one view and one theme active to run this function!","")
   exit
else
   ans = MsgBox.YesNo("Do you want to select an area in the  "+theThemeName.AsString+
   " theme of the "+theView.AsString+" view?","Are you sure?", true)
   if (ans = false) then
```

```
        exit
    end
end

'  Find the "mark" and "area_pts" fields in the table
_markField = theVTab.FindField("mark")
_areaField = theVTab.FindField("area_pts")

'  Ensure that the values under the two fields are set to zero
'  If not, INITIALIZE!
if (theVTab.StartEditingWithRecovery) then
    for each rec in theVTab
        if (theVTab.ReturnValue(_markField, rec)<>0) then
            theVTab.BeginTransaction
            theVTab.SetValue(_markField, rec, 0)
            theVTab.EndTransaction
        end
    end

    for each rec in theVTab
        if (theVTab.ReturnValue(_areaField, rec)<>0) then
            theVTab.BeginTransaction
            theVTab.SetValue(_areaField, rec, 0)
            theVTab.EndTransaction
        end
    end
else
    MsgBox.Warning("Problem Editing Table!","")
    exit
end

'  Allow user to select entire data set or a portion of it
optionList = {"...select the entire data set?", "...select a portion of data?"}
option = MsgBox.ChoiceAsString(optionList, "Do you want to...", "Pick An Option")
if (option = nil) then
    theVTab.StopEditingWithRecovery(false)
    exit
elseif (option = optionList.Get(0)) then
    theBitmap.SetAll
    theVTab.UpdateSelection
    _portion = false
else
    for each t in av.GetActiveGUI.GetToolBar
```

```
        if ((t.Is(Tool)) and (t.GetApply = "View.SelectPoint")) then
            t.Select
            break
        end
    end
    ' If "only a portion of the data set" is being selected...This flag will
    ' allow the selected portion to be added to the view as a new theme.
    _portion = true
end

theVTab.StopEditingWithRecovery(false)
*******************************************

*******************************************
'Name:  Proj.startup
'Author/Date:  Pam LaFontaine / 18 April 98
'Updated_by/Date:  16 Jan 99
'Course:          URP6272, Spring 1998
'Prepared at:     GeoPlan Center, University of Florida
'---------------------------------------------------
'Description:  Startup script for the SHOALS Surface Simplification
'              algorithm which will take a set of data depths and
'              intelligently "thin them out" into a smaller set.
'Requires:
'Calls:
'Is called by:
'Self:
'Returns:
'FileName:  Proj.startup
'---------------------------------------------------

' Gives the ArcView window a new title.
av.SetName ("SHOALS Final Draft Surface Simplification - Summer 1999")

' Gives the project window a title
av.GetProject.SetName ("Pam LaFontaine-Thesis")

' Flash SHOALS image banner

MsgBox.Banner("shoal.gif".AsFileName, 7, "Pam LaFontaine's Application For..."++
              "(for more information contact plafon@ufl.edu)")

' Initialize variable used in the Proj.convert script
```

```
'_shapeFromScript = false

'  Initialize variable used to determine if the user has selected only a portion of
'  the data set (see Proj.select)
_portion = false
******************************************

******************************************
'Name:  Proj.stats_calc
'Author/Date:  Pam LaFontaine / 4 Aug 98
'Updated_by/Date:  4 July 99
'Course:          URP6272, Spring 1998
'Prepared at:      GeoPlan Center, University of Florida and NIMA
'--------------------------------------------------
'Description:  Performs surface/grid operations and creates a statistics table
'Requires:
'Calls:
'Is called by:  Proj.statistics
'Self:  denseSurf = SELF.Get(0), sparseSurf = SELF.Get(1)
'Returns:  statsList = {statsSurfVTab, sumSurfVTab}
'FileName:  Proj.stats_calc
'--------------------------------------------------

theView = av.GetActiveDoc

'  Identify all of the surfaces being passed from the "Proj.statistics" script
denseSurf = SELF.Get(0)
sparseSurf = SELF.Get(1)

'  ******************************************

'  Make a Zone Grid (Integer Grid) to define which cells will be used in calculations
'  In actuallity all cells of each surface grid will be used (there should be no null
'  values if the interpolation process was effective)
zoneGrid = (denseSurf.IsNull).Con(0.AsGrid, 1.AsGrid)

'  ******************************************

'  Subtract the sparse surface from the dense surface ...there is a difference
'  value for each cell location
surfDiff = denseSurf - sparseSurf

'  ******************************************
```

```
' Square the values that make up surfDiff (We will add these values together
' and divide by the total number of cells involved)
surfDiffSqr = surfDiff * surfDiff

' ******************************************

' Prj.MakeNull will be used for the zone projection to default to the
' current view's projection

' Identify the Zone Field for the zoneGrid
zoneGridTheme = GTheme.Make(zoneGrid)
zoneVTab = zoneGridTheme.GetVTab
if (zoneVTab = nil) then
   MsgBox.Info("Could not get the zoneVTab!","")
   exit
end
zoneField = zoneVTab.FindField("Value")

' (No Data = true) means that the statistics will not be computed for
' cells with a value of No Data (All of our cells should contain data anyway!)

' Create a filename for the zone statistics
zoneFN = av.GetProject.GetWorkDir.MakeTmp("zone","dbf")

' ******************************************

' Create the Zonal Statistics Table for surfDiff values
statsSurfVTab = (surfDiff).ZonalStatsTable(zoneGrid, Prj.MakeNull, zoneField, true, zoneFN)

' Check if problems creating the statsSurfVTab
if (statsSurfVTab.HasError) then
   MsgBox.Error("The statsSurfVTab has an error!","")
   exit
end

'
************************************************************************

' Find the sum of the squares of "denseSurf - sparseSurf" (or surfDiff) residuals
sumSurfDiffSqr = (surfDiffSqr).ZonalStats(#GRID_STATYPE_SUM, zoneGrid,
Prj.MakeNull, zoneField, true)
```

```
if (sumSurfDiffSqr.HasError) then
   MsgBox.Error("The sumSurfDiffSqr had an error!","")
   exit
end

'  Retrieve the sum value stored in each cell, without loosing precision of the number
'  (We will divide by 100 later to account for this step)
multSumSurfDiffSqr = (sumSurfDiffSqr * 100.AsGrid).Int

sumSurfDiffSqrVTab = multSumSurfDiffSqr.GetVTab
if (sumSurfDiffSqrVTab = nil) then
   MsgBox.Info("Could not get the sumSurfDiffSqrVTab!","")
   exit
end

'
**********************************************************************

'  Find the number of cells in the surfDiffSqr grid.
'  Find number of rows and number of columns and multiply
rcList = surfDiffSqr.GetNumRowsAndCols
numRows = rcList.Get(0)
'MsgBox.Info("The number of rows is "+numRows.AsString,"Rows")
numCols = rcList.Get(1)
'MsgBox.Info("The number of cols is "+numCols.AsString,"Cols")
numCells = numRows * numCols
'MsgBox.Info("The number of cells is "+numCells.AsString,"Total number of Cells")

'
**********************************************************************

'  As a check on the value computed for the standard deviation in the ZonalStats process
'  also compute the sum of the surfDiff variable.  This sum will be divided by the number
'  of data points to find the mean surfDiff value and then squared in order to calculate
'  the standard deviation from the RMS value.

'  Find the sum of the absolute values of the residuals "densSurf - sparseSurf" or "surfDiff"
'  NOTE:  I DO NOT WANT TO TAKE THE ABS. VALUE HERE.  I WANT TO FIND
' THE AVERAGE VALUE WHICH ACCOUNTS FOR THE SURFACES DIFFERING
' IN BOTH THE POSITIVE AND NEGATIVE DIRECTIONS!!!
sumSurfDiff = (surfDiff).ZonalStats(#GRID_STATYPE_SUM, zoneGrid, Prj.MakeNull,
zoneField, true)
```

```
if (sumSurfDiff.HasError) then
   MsgBox.Error("The sumSurfDiff had an error!","")
   exit
end

'  Retrieve the sum value stored in each cell, without loosing precision of the number
'  (We will divide by 1,000 later to account for this step)
multSumSurfDiff = (sumSurfDiff * 1000.AsGrid).Int

sumSurfDiffVTab = multSumSurfDiff.GetVTab
if (sumSurfDiffVTab = nil) then
   MsgBox.Info("Could not get the sumSurfDiffVTab!","")
   exit
end

'
*************************************************************************

statsList = {statsSurfVTab, sumSurfDiffSqrVTab, sumSurfDiffVTab, numCells}

return statsList
*******************************************

*******************************************
'Name:  Proj.statistics
'Author/Date:  Pam LaFontaine / 4 Aug 98
'Updated_by/Date:  25 Aug 99
'Course:           URP6272, Spring 1998
'Prepared at:      GeoPlan Center, University of Florida and NIMA
'---------------------------------------------------
'Description:  Computes a table of statistics to compare how surfaces fit to two point sets
'Requires:
'Calls:  Proj.make_surface, and Proj.stats_calc
'Is called by:  (Menu request -- Run Statistics)
'Self:
'Returns:
'FileName:  Proj.statistics
'---------------------------------------------------

'  Get a file name and path for your new table
aFileName = FileDialog.Put("Comparison_Table.dbf".AsFileName, "*.dbf", "Create the
Comparison Table")
if (nil = aFileName) then
```

```
   exit
end

'  Create a dBase table to hold results of comparison runs
finalVTab = VTab.MakeNew (aFileName, dBase)

finalVTab.SetEditable(true)
runField = Field.Make("Run No", #FIELD_SHORT, 4, 0)
radField = Field.Make("radius", #FIELD_FLOAT, 6, 2)
tolField = Field.Make("tolerance", #FIELD_FLOAT, 6, 2)
denseField = Field.Make("dense", #FIELD_LONG, 10, 0)
thinField = Field.Make("thin", #FIELD_LONG, 10, 0)
RMSsurfField = Field.Make("surf RMS", #FIELD_FLOAT, 10, 6)
MinDiffField = Field.Make("Min Diff", #FIELD_FLOAT, 8, 3)
MaxDiffField = Field.Make("Max Diff", #FIELD_FLOAT, 8, 3)
StdDevField = Field.Make ("Std Dev", #FIELD_FLOAT, 10, 6)
ManStdDevField = Field.Make ("Manual Std Dev", #FIELD_FLOAT, 10, 6)
numField = Field.Make ("N", #FIELD_LONG, 10, 0)
meanField = Field.Make ("mean", #FIELD_FLOAT, 10, 4)

finalVTab.AddFields({runField, radField, tolField, denseField, thinField,
RMSsurfField, MinDiffField,
                MaxDiffField, StdDevField, ManStdDevField, numField, meanField})

'
*********************************************************************

'  Allows you to go back out to the menu window to set the Analysis Properties to
'  match the Area of Interest!  If this is not done, your process will BOMB!
ans = MsgBox.YesNo("Have you set your Analysis Properties to correspond with the AOI?  ",
    "First Set Analysis Properties", true)
if (ans = false) then
   exit
end

MsgBox.Report("Select two point coverage themes you wish to compare, one dense and"++
          "the other more sparse.  (Hold down the shift key to select two themes.)"+NL+
          "The themes must be point shapefiles, and not grids or images!",
          "Select Two Themes To Compare")

go = TRUE
run = 1
while (go)
```

```
runValue = run

theView = av.GetActiveDoc
activeThemes = theView.GetActiveThemes
activeThemeCount = theView.GetActiveThemes.Count

if (activeThemeCount = nil) then
   exit
elseif (activeThemeCount <> 2) then
   MsgBox.Info("You must have TWO shape themes selected to make this statictical"++
         "comparison", "")
   exit
elseif ((activeThemes.Get(0).Is(Ftheme).Not) OR (activeThemes.Get(1).Is(Ftheme).Not))
then
   MsgBox.Info("You must be working with shapefile themes and not grids or images!", "")
   exit
else
   theTheme1 = activeThemes.Get(0)
   theFTab1 = theTheme1.GetFTab
   theTheme2 = activeThemes.Get(1)
   theFTab2 = theTheme2.GetFTab
   FTab1count = theFTab1.GetNumRecords
   FTab2count = theFTab2.GetNumRecords
   if (FTab1count > FTab2count) then
      _denseValue = FTab1count
      _thinValue = FTab2count
      '  Assuming Analysis Properties Are Set for the project
      denseSurfaceGridThm = av.Run("Proj.make_surface", theTheme1)
      denseSurf = denseSurfaceGridThm.GetGrid
      sparseSurfaceGridThm = av.Run("Proj.make_surface", theTheme2)
      sparseSurf = sparseSurfaceGridThm.GetGrid
   else
      _denseValue = FTab2count
      _thinValue = FTab1count
      '  Assuming Analysis Properties Are Set for the project
      denseSurfaceGridThm = av.Run("Proj.make_surface", theTheme2)
      denseSurf = denseSurfaceGridThm.GetGrid
      sparseSurfaceGridThm = av.Run("Proj.make_surface", theTheme1)
      sparseSurf = sparseSurfaceGridThm.GetGrid

   end
end
```

```
surfList = {denseSurf, sparseSurf}

statsList = av.Run("Proj.stats_calc", surfList)

statsSurfVTab = statsList.Get(0)
sumSurfDiffSqrVTab = statsList.Get(1)
sumSurfDiffVTab = statsList.Get(2)
numCells = statsList.Get(3)

' *********************************************
' Identify the "value" field of the individual cells in the sumSurfDiffSqrVTab grid
sumFieldSurfDiffSqr = sumSurfDiffSqrVTab.FindField("Value")

' Retrieve the value which is in each of the cells of the sumSurfDiffSqrVTab grid
multSumSurfDiffSqr = sumSurfDiffSqrVTab.ReturnValue(sumFieldSurfDiffSqr, 0)

' Divide multSumSurfDiffSqr by 100, since earlier (in Proj.stats_calc) we multiplied
' the number by this value in order to retain precision
sumSurfDiffSqr = multSumSurfDiffSqr/100

' *********************************************
' Identify the "value" field of the individual cells in the sumSurfDiffVTab grid
sumFieldSurfDiff = sumSurfDiffVTab.FindField("Value")

' Retrieve the value which is in each of the cells of the sumSurfDiffVTab grid
multSumSurfDiff = sumSurfDiffVTab.ReturnValue(sumFieldSurfDiff, 0)

' Divide multSumSurfDiff by 1000, since earlier (in Proj.stats_calc) we multiplied
' the number by this value in order to retain precision
sumSurfDiff = multSumSurfDiff/1000

' Now to complete the manual computation (for comparison purposes) of the Standard
Deviation of
' the "surfDiff" grid values, find the mean of the sumSurfDiff value and then square it

mean = sumSurfDiff/numCells
meansqr = mean * mean

' ...Finally subtract this "meansqr" value from the "sumSurfDiffSqr/numCells" value and take
' the square root of the whole thing... This turns your RMS into a Standard Deviation!!

manualStdDev = (sumSurfDiffSqr/numCells-meansqr).Sqrt
```

```
   ' *********************************************

   minimumField = statsSurfVTab.FindField("Min")
   minimum = statsSurfVTab.ReturnValue(minimumField, 0)
   maximumField = statsSurfVTab.FindField("Max")
   maximum = statsSurfVTab.ReturnValue(maximumField, 0)
   SDField = statsSurfVTab.FindField("Std")
   sd = statsSurfVTab.ReturnValue(SDField, 0)

   ' *******************************************************************

   '  RMS (root mean square) is computed across all cells of the difference surface
   RMSsurf = (sumSurfDiffSqr/numCells).Sqrt
   'MsgBox.Info("The RMS is "+RMSsurf.AsString,"Root Mean Squared of Two Surfaces")

   ' *********************************************

   rec = finalVTab.AddRecord
   finalVTab.SetValue(runField, rec, runValue)
   finalVTab.SetValue(radField, rec, _dist)
   finalVTab.SetValue(tolField, rec, _tol)
   finalVTab.SetValue(denseField, rec, _denseValue)
   finalVTab.SetValue(thinField, rec, _thinValue)
   finalVTab.SetValue(RMSsurfField, rec, RMSsurf)
   finalVTab.SetValue(MinDiffField, rec, minimum)
   finalVTab.SetValue(MaxDiffField, rec, maximum)
   finalVTab.SetValue(StdDevField, rec, sd)
   finalVTab.SetValue(ManStdDevField, rec, manualStdDev)
   finalVTab.SetValue(numField, rec, numCells)
   finalVTab.SetValue(meanField, rec, mean)

   ' *********************************************
go = FALSE

end

finalVTab.SetEditable(false)
******************************************

******************************************
'Name:  Proj.thin
'Author/Date:  Pam LaFontaine / 22 April 98
'Updated_by/Date:  24 Aug 99
```

```
'               Added the Z-value/Continue code....
'Course:        URP6272, Spring 1998
'Prepared at:   GeoPlan Center, University of Florida
'---------------------------------------------------
'Description:  Gets the VTab associated with the view's active theme.
'              Goes through each record of the attribute table and
'              tests whether or not the record should be "marked".
'              Records are "marked" if the z-value of the operating
'              point (Zop) falls within the user-established tolerance
'              of the z-value of the test point (Ztest).
'Requires:
'Calls:
'Is called by:  Proj.make_area_1s and Proj.shapefile
'Self:  theVTab = SELF.Get(0), theTheme = SELF.Get(1)
'Returns:  theVTab
'FileName:  Proj.thin
'---------------------------------------------------

'  Assign the SELF variables
theVTab = SELF.Get(0)
theTheme = SELF.Get(1)

'  Find the "mark", "area_pts" and "Z" fields in the table
_markField = theVTab.FindField("mark")
_areaField = theVTab.FindField("area_pts")
_zField = theVTab.FindField("Z")

'  User can determine the radius of the circle used to select the points for analysis
dist = MsgBox.Input("Enter the desired search radius in feet"+NL+"(Value Range: 0-50):
","Neighborhood of Analysis","")
'  Set a global variable to be passed to the table in the statistics script later
_dist = dist
if ((dist = nil) OR (dist < 0)) then
   exit
end

'  User can determine the tolerance within which a z-value (depth) will be rejected.
'  Tolerance is the difference value between two depths which is determined
'  to make the least shoal of the two depths redundant.  The redundant depths are
'  then "marked" in a separate field and will not be included in the output
tolerance = MsgBox.Input("Enter the tolerance in feet (Value Range: 0-1):  ","Analysis
Tolerance","")
'  Set a global variable to be passed to the table in the statistics script later
```

```
_tol = tolerance
if ((tolerance = nil) OR (tolerance < 0)) then
  exit
end

' Go through each record in selected area and perform analysis
for each index in theVTab
  area = theVTab.ReturnValue(_areaField, index)
  if (area = 0) then
    continue
  else
    mark = theVTab.ReturnValue(_markField, index)
    if (mark = 1) then
      continue
    else
      if (theVTab.StartEditingWithRecovery) then
        theBitmap = theVTab.GetSelection
        theBitmap.Clearall
        theVTab.UpdateSelection
        ' Put the point itself into bitmap set
        theBitmap.Set(index)
        theVTab.UpdateSelection
        ' Add to bitmap set the points within radius

theTheme.SelectByTheme(theTheme,#FTAB_RELTYPE_ISWITHINDISTANCEOF,
                          dist.AsNumber,#VTAB_SELTYPE_OR)
        theVtab.UpdateSelection
        Zop = theVTab.ReturnValue(_zField, index)

        '**********************************************************
        ' Marking process is shoal-biased...the most shoal (or least deep)
        ' of the two points is kept in the new set!
        for each rec in theVTab.GetSelection
          ' Do not make the Z-value comparison with points already marked for removal
          mark1 = theVTab.ReturnValue(_markField, rec)
          if (mark1 = 1) then
            continue
          elseif (rec <> index) then
            Ztest = theVTab.ReturnValue(_zField, rec)
            diff = Zop-Ztest
            if (diff.Abs < tolerance.AsNumber) then
              if (Zop < Ztest) then
                theVTab.BeginTransaction
```

```
                    ' mark the operating value's Z-field
                    theVTab.SetValue(_markField, index, 1)
                    theVTab.EndTransaction
                  else
                    theVTab.BeginTransaction
                    ' mark the test value's Z-field
                    theVTab.SetValue(_markField, rec, 1)
                    theVTab.EndTransaction
                  end
                else
                  continue
                end
              else
                continue
              end
            end
            ' ***********************************************************
          else
            MsgBox.Warning("Problem Editing Table!","")
            exit
          end
        end
      end
    end

    return theVTab
    *******************************************
    *******************************************
```

APPENDIX B
FOURIER CONCEPTS AND EQUATIONS

The fluctuations in a section of terrain can be mathematically modeled by sine and cosine waves. Each cross-section of the terrain could be described as a complicated wave; and a complicated wave, no matter how irregular, is a combination of sine and cosine waves of various frequencies and amplitudes. Using the Fourier series formula, a complicated periodic wave can be expressed as the sum of simple sine and cosine waves:

$$\begin{aligned} f(t) = a_0 &+ a_1 \cos wt + b_1 \sin wt \\ &+ a_2 \cos 2wt + b_2 \sin 2wt \\ &+ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &+ a_n \cos nwt + b_n \sin nwt \end{aligned}$$

The a's and b's (with subscripts) are the amplitudes, or quantities, of each wave, with a_0 representing the amount by which the entire wave is shifted up or down along the y-axis (amplitude shift).

The Fourier series formula can be more succinctly expressed using summation symbology, with the angular velocity expressed in either degrees or radians. In each case, the frequency (f) is the reciprocal of the period (T).

For ω(angular velocity) in degrees:

$$f(t) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos nwt + b_n \sin nwt \right)$$

where $w = 360 \cdot f = 360 \cdot \frac{1}{T}$, since $f = \frac{1}{T}$

For 2π/T(angular velocity) in radians:

$$f(t) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{2p\,n}{T} t + b_n \sin \frac{2p\,n}{T} t \right)$$

where $w = 2p\,f = \frac{2p}{T}$, since $f = \frac{1}{T}$

In the formats above, nwt or $\frac{2p\,n}{T}t$ is a multiple of the fundamental frequency of each complicated wave being described. Regardless of the format, the Fourier series is considered the "ultimate wave formula".

By using filters and integration, the amplitude of each simple wave may be determined. This is the equivalent of finding out how much of each wave exists, or the contribution of each wave to the terrain as a whole. The formulae for a_0, a_n, and b_n, where $f(t)$ represents a complicated wave, are called the Fourier coefficients:

$$a_0 = \frac{1}{T}\int_0^T f(t)dt$$

$$a_n = \frac{2}{T}\int_0^T f(t)\cos n\omega\, t\, dt$$

$$b_n = \frac{2}{T}\int_0^T f(t)\sin n\omega\, t\, dt$$

In accordance with Euler's formula:

$$e^{i\theta} = \cos\theta + i\sin\theta \qquad \text{where } e = \lim_{t\to 0}(1+t)^{\frac{1}{t}} = 2.71828182 \text{ and } i^2 = -1$$

More exactly, Euler's formula can be broken down to express θ as $-\theta$ and as $+\theta$.

For a positive angle: $e^{i\theta} = \cos\theta + i\sin\theta$

And, for a negative angle: $e^{-i\theta} = \cos\theta - i\sin\theta$

Now, solving for $\cos\theta$ and $\sin\theta$: $\cos\theta = \dfrac{e^{i\theta} + e^{-i\theta}}{2}$ and $\sin\theta = \dfrac{e^{i\theta} - e^{-i\theta}}{2}$

With $\theta = n\omega t$, Euler's formula may be expressed as follows:

$$e^{in\omega t} = \cos n\omega t + i\sin n\omega t \qquad \text{and} \qquad e^{-in\omega t} = \cos n\omega t - i\sin n\omega t$$

Then, solving for the cosine and sine of the angle:

$$\cos n\omega t = \frac{1}{2}\left(e^{in\omega t} + e^{-in\omega t}\right) \qquad \text{and} \qquad \sin n\omega t = \frac{1}{2i}\left(e^{in\omega t} - e^{-in\omega t}\right)$$

Therefore, the Fourier series may be rewritten as:

$$f(t) = a_0 + \sum_{n=1}^{\infty}\left\{\frac{a_n}{2}\left(e^{in\omega t} + e^{-in\omega t}\right) + \frac{b_n}{2i}\left(e^{in\omega t} - e^{-in\omega t}\right)\right\}$$

where the Fourier coefficients are:

$$a_0 = \frac{1}{T}\int_0^T f(t)dt$$

$$a_n = \frac{2}{T}\int_0^T f(t)\frac{1}{2}\left(e^{in\omega t} + e^{-in\omega t}\right)dt$$
$$b_n = \frac{2}{T}\int_0^T f(t)\frac{1}{2i}\left(e^{in\omega t} - e^{-in\omega t}\right)dt$$

Simplifying terms we have the Fourier series formula:

$$f(t) = a_0 + \sum_{n=1}^{\infty}\left\{\frac{1}{2}(a_n - ib_n)e^{in\omega t} + \frac{1}{2}(a_n + ib_n)e^{-in\omega t}\right\}$$

And the Fourier coefficients:

$$a_0 = \frac{1}{T}\int_0^T f(t)dt$$
$$a_n = \frac{1}{T}\int_0^T \left\{ f(t)e^{in\omega t} + f(t)e^{-in\omega t}\right\}dt$$
$$b_n = \frac{1}{Ti}\int_0^T \left\{ f(t)e^{in\omega t} - f(t)e^{-in\omega t}\right\}dt$$

Removing the i from the right side of the b_n formula, the formulae for a_n and b_n look very similar. Each can be simplified to a three term equation as indicated below:

$$a_n = \frac{1}{T}\int_0^T f(t)e^{in\omega t}dt + \frac{1}{T}\int_0^T f(t)e^{-in\omega t}dt \qquad \text{or} \qquad A = X + Y$$

and

$$ib_n = \frac{1}{T}\int_0^T f(t)e^{in\omega t}dt - \frac{1}{T}\int_0^T f(t)e^{-in\omega t}dt \qquad \text{or} \qquad B = X - Y$$

Using simultaneous equations, X and Y may be determined:

$$X = \frac{1}{2}(A + B) \qquad \text{and} \qquad Y = \frac{1}{2}(A - B)$$

Now, with substitution, the equations become:

$$\frac{1}{2}(a_n - ib_n) = \frac{1}{T}\int_0^T f(t)e^{-in\omega t}dt \qquad \text{and} \qquad \frac{1}{2}(a_n + ib_n) = \frac{1}{T}\int_0^T f(t)e^{in\omega t}dt$$

If we set $\frac{1}{2}(a_n - ib_n) = A_n$ and $\frac{1}{2}(a_n + ib_n) = B_n$, then the Fourier series formula looks like this:

$$f(t) = a_0 + \sum_{n=1}^{\infty}\left\{ A_n e^{in\omega t} + B_n e^{-in\omega t}\right\}$$

where

$$A_n = \frac{1}{T}\int_0^T f(t)e^{-in\omega t}dt \qquad \text{and} \qquad B_n = \frac{1}{T}\int_0^T f(t)e^{in\omega t}dt$$

A final simplification expands the range for $\sum$ and includes a_0, A_n and B_n in a new coefficient called C_n. The complex number representation of Fourier series becomes simply:

$$f(t) = \sum_{n=-\infty}^{\infty} C_n e^{in\omega t}$$

and the complex number representation of Fourier coefficients becomes:

$$C_n = \frac{1}{T}\int_0^T f(t)e^{-in\omega t}dt$$

The Fourier transform is used to describe <u>non-periodic</u> waves. Here the period is expressed in terms of the fundamental frequency:

$$\Delta f\,(\text{fundamental frequency}) = \frac{1}{T(\text{period})}$$

If $T \to \infty$, as it would in a wave that does not repeat itself, then $\Delta f = \frac{1}{T} \to \frac{1}{\infty}$. As the fundamental frequency approaches zero, the wave becomes more continuous. All frequencies would be defined only if the period could reach infinity.

We can express the Fourier coefficients formula as:

1. $$C_n = \Delta f \int_{-T/2}^{T/2} g(t)e^{-i2\pi f_n t}dt \qquad \text{where} \quad \omega = 2\pi f \quad \text{and} \quad n\omega = 2\pi f_n$$

and the Fourier series as:

2. $$g(t) = \sum_{n=-\infty}^{\infty} C_n e^{i2\pi f_n t} \qquad \text{where} \quad \omega = 2\pi f \quad \text{and} \quad n\omega = 2\pi f_n$$

Now replace C_n in Equation 2 with the formula for C_n in Equation 1. This gives us:

$$g(t) = \sum_{n=-\infty}^{\infty} \left\{ \Delta f \int_{-T/2}^{T/2} g(t)e^{-i2\pi f_n t}dt \right\} e^{i2\pi f_n t}$$

If we take the limit as T goes towards infinity, the formula can be written like this:

$$g(t) = \int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\infty} g(t)e^{-i2\pi f t}dt \right\} e^{i2\pi f t}df$$

When $T \to \infty$, the values for T become continuous, as do the values for f. That is why f_n loses its subscript n and the formula no longer jumps from frequency to frequency. This double integral formula is expressed as a function of frequency (f). It is actually the Fourier coefficients formula written as a function of f. The equation can be further simplified and expressed as the Fourier transform:

$$G(f) = \int_{-\infty}^{\infty} g(t)\, e^{-i2\pi f t}\, dt$$

It is the formula for the Fourier coefficients of any wave, including non-periodic waves.

The Fourier series formula can be re-written as well. In this format it is known as the Inverse Fourier transform:

$$g(t) = \int_{-\infty}^{\infty} G(t)\, e^{i2\pi f t}\, df$$

The Fourier transform does not provide the actual amplitudes of the component waves, but it does indicate the "relationship" among their amplitudes, or the ratio of the various amplitudes.

This knowledge of the relative contribution of each simple wave component is sufficient for wave analysis. Using the Fourier transform, a wave can be observed only over a finite period of time, Δt. Nothing is known about the part of the wave beyond the window of time over which it is observed. This lack of precise knowledge about the characteristics of the entire wave is called "uncertainty". The Fourier transform can be used to calculate $G(f)$ over a period of time Δt by changing the form of the equation to:

$$G(f) = \frac{\sin \pi f \Delta t}{\pi f \Delta t} \Delta t \quad \text{where } G(f) = \Delta t \text{ when } f = 0.$$

In order to find individual values for the Fourier transform formula over time, values are calculated at regular intervals, τ. That means $g(t) = g(k\tau)$ where k represents the interval number, or the number of a given value in a series:

$$G\left(\frac{n}{\tau N}\right) = \sum_{k=0}^{N-1} \left\{ \tau \cdot g(k\tau) e^{-i2\pi k \frac{n}{N}} \right\} \qquad \text{where } f = \frac{n}{\tau N},\ N = \text{total number of values},$$

τ = interval between values, k = place order of each value, and
n = integral multiple of fundamental frequency.

By changing τ to a value of one, the formula becomes:

$$G\left(\frac{n}{N}\right) = \sum_{k=0}^{N-1} g(k) e^{-i2\pi k \frac{n}{N}}$$

If we let $e^{\frac{-i2\pi}{N}} = W$, then $G\left(\frac{n}{N}\right) = \sum_{k=0}^{N-1} g(k) W^{nk}$.

Now if the values for k are divided into odd and even values, the number of calculations can be reduced significantly:

$$G\left(\frac{n}{N}\right) = \sum_{k=0}^{N-1} g(k) W^{nk} = \sum_{k=0}^{\frac{N}{2}-1} g(2k) W^{n2k} + \sum_{k=0}^{\frac{N}{2}-1} g(2k+1) W^{n(2k+1)}$$

where Even numbers $\rightarrow 2k$ $(k = 0,1,2,3\ldots)$
Odd numbers $\rightarrow 2k+1$ $(k = 0,1,2,3\ldots)$

This formula can be simplified by allowing the:
Even group = $g(2k) = p(k)$
Odd group = $g(2k+1) = q(k)$

Then $G\left(\dfrac{n}{N}\right) = \displaystyle\sum_{k=0}^{\frac{N}{2}-1} p(k)W^{2nk} + W^{n}\sum_{k=0}^{\frac{N}{2}-1} q(k)W^{2nk}$.

The Fast Fourier transform (FFT) uses an N value of 2^n in order to speed up the process by reducing the number of calculations required. The value W is called a rotation operator and allows for this simplified processing and the faster calculation of wave spectra.

Using the FFT, the individual components of a complex wave can be quickly computed. The component sine and cosine waves may differ in frequency and in their ratio of contribution to the complicated wave. Added together, the component waves represent the contours of a cross-section of terrain. Using a two-dimensional FFT, the ratio of contribution for each component wave frequency could be calculated. The lower frequencies comprise the lower harmonics and the higher frequencies make up the higher harmonics.

If a section of terrain is primarily made up of waves in the low harmonics (low frequencies), then a larger radius value should suffice in the data density reduction algorithm. This is because the terrain made up of lower frequencies is relatively flat, or without a lot of variation in the elevation values. A large radius could probably be used without significant loss of detail in the reduced DTM generated.

If the terrain is found to contain large quantities of high frequency component waves, it is more irregular in nature, with a lot of change in the z-values of the DTM. Only by using a smaller radius value could one be sure to capture the detail of such variable terrain. The DDR algorithm works more slowly with a smaller radius parameter input, but tends to generate a more statistically accurate DTM.

This explanation of Fourier concepts was derived from the book “Who is Fourier: A Mathematical Adventure” as written by the Transnational College of LEX in 1997.

REFERENCES

Bertrand, G., 1995. "A Parallel Thinning Algorithm for Medial Surfaces," Pattern Recognition Letters, Vol. 16, No. 9, 979-986.

Bertrand, G., 1994. "Simple Points, Topological Numbers, and Geodesic Neighborhoods in Cubic Grids," Pattern Recognition Letters, Vol. 15, No. 10, 1003-1011.

Brady, M.D. and D.R. Ford, 1990. Intelligent Data Reduction (IDARE) NAG8-642 Final Report, National Aeronautics and Space Administration George C. Marshall Space Flight Center, Huntsville, Alabama, Microfiche.

Deutsch, E.S., 1972. "Thinning Algorithms in Rectangular, Hexagonal, and Triangular Arrays," Communications of the ACM, Vol. 15, No. 9, 827-837.

Estep, L.L., W.J. Lillycrop, and L.E. Parson 1994. "Estimation of Maximum Depth of Penetration of a Bathymetric Lidar System Using a Secchi Depth Database," MTS Journal, Vol. 28, No. 2, 30-36.

Guenther, G.C., 1985. "Airborne Laser Hydrography: System Design and Performance Factors," NOAA Professional Paper Series, National Ocean Service 1, National Oceanic and Atmospheric Administration, Rockville, MD, 385 p.

Guenther, G.C., T.J. Eisler, J.L. Riley, and S.W. Perez, 1996. "Obstruction Detection and Data Decimation for Airborne Laser Hydrography," Proc. 1996 Canadian Hydrographic Conference, June 3-5, 1996, Halifax, Nova Scotia.

Guenther, G.C., P.E. LaRocque, and W.J. Lillycrop, 1994. "Multiple Surface Channels in SHOALS Airborne Lidar," Proc. SPIE: Ocean Optics XII, Vol. 2258, 422-430.

Guenther, G.C., R.W.L. Thomas, and P.E. LaRocque, 1996. "Design Considerations for Achieving High Accuracy with the SHOALS Bathymetric Lidar System," SPIE: Laser Remote Sensing of Natural Waters from Theory to Practice, Vol. 2964, 54-71.

Irish, J.L., W.J. Lillycrop, and L.E. Parson, 1996. "Accuracy of Sand Volumes as a Function of Survey Density," Proc. 25th International Conference on Coastal Engineering, September 2-6, 1996, Orlando, Florida.

Irish, J.L., W.J. Lillycrop, L.E. Parson, and M.W. Brooks, 1994. "SHOALS System Capabilities for Hydrographic Surveying," Dredging '94: Proc. Second International Conference on Dredging and Dredged Material Placement, American Society of Civil Engineers, November 13-16, 1994, Orlando, Florida, 314-321.

Irish, J.L., J.E. Thomas, L.E. Parson, and W.J. Lillycrop, 1996. "Monitoring Storm Response With High Density Lidar Bathymetry: The Effects of Hurricane Opal on Florida's Panhandle," Proc. Second International Airborne Remote Sensing Conference and Exhibition, Environmental Research Institute of Michigan, June 24-27, 1996, San Francisco, California.

Li, S.Z., 1993. "Self-Organization of Surface Shapes," Proc. 1993 International Conference on Neural Networks, Vol. 2, October 25-29, 1993, Nagoya, Japan.

Lillycrop, W.J. and L.L. Estep, 1995. "Generational Advancements in Coastal Surveying, Mapping: In U.S.-Canada Project, U.S. Army Corps of Engineers Merges Airborne Lidar, Hyperspectral Imagers, GIS, DGPS, Spurring Commercialization," Sea Technology, Vol. 36, No. 6, 10-16.

Lillycrop, W.J., J.L. Irish, and L.E. Parson, 1997. "SHOALS System: Three Years of Operation with Airborne Lidar Bathymetry--Experiences; Capability and Technology Advancements," Sea Technology, Vol. 38, No. 6, 17-25.

Lillycrop, W.J., L.E. Parson, and J.L. Irish, 1996. "Development and Operation of the SHOALS Airborne Lidar Hydrographic Survey System," SPIE: Laser Remote Sensing of Natural Waters from Theory to Practice, Vol. 2964, 26-37.

Parson, L.E., W.J. Lillycrop, C.J. Klein, R.C.P. Ives, and S.P. Orlando, 1997. "Use of Lidar Technology for Collecting Shallow Water Bathymetry of Florida Bay," Journal of Coastal Research, Vol. 13, No. 4, 1173-1180.

Press, W.H., S.A. Teukolsky, W.T. Vetterling, and B.P.Flannery, 1992. *Numerical Recipes in C: The Art of Scientific Computing, 2nd ed.* (Cambridge, United Kingdom: Cambridge University Press).

Transnational College of LEX, 1997. W*ho is Fourier?: A Mathematical Adventure* (Boston, MA: Language Research Foundation)

Watters, T. and C.E. Wiggins, 1999. “Utilization of Remote Sensing Methods for Management of Florida’s Coastal Zone,” Proc. 1999 U.S. Hydrographic Conference, April 26-29, Mobile, AL.

Weber, W., 1977. "Optimal Approximation in Automated Cartography," Optimal Estimation in Approximation Theory, C.A. Micchelli and T.J. Rivlin, Eds., Plenum Press, New York.

BIOGRAPHICAL SKETCH

Pamela Sue (Finley) LaFontaine was born in Lowell, Massachusetts on June 17, 1966. Until the fifth grade, she lived and attended school primarily in Massachusetts or Ohio. In 1976, she and her family moved to the Washington, D.C. metropolitan area where she completed her pre-college education. She graduated from Friendly Senior High in 1984, and entered the U.S. Naval Academy in Annapolis, Maryland later that summer. In 1988, she graduated from the Naval Academy with a Bachelor of Science degree in Oceanography. The following five years were dedicated to Naval Service, with Mediterranean and Persian Gulf tours on board the ship USS SAN DIEGO (AFS-6), and a final tour at a shore command in Chinhae, Korea. After a short-term position at the Virginia Institute of Marine Science, the former Ms. Finley found employment as a cartographer and marine information specialist at the former Defense Mapping Agency (DMA), now called the National Imagery and Mapping Agency (NIMA). It was at this agency that she also found her husband, Donald LaFontaine. They were married in August of 1996. Soon thereafter, Ms. LaFontaine was selected by NIMA for their Long-Term Full-Time Training program. In August of 1997, she moved to Gainesville to pursue a master's degree in the Geomatics program at the University of Florida.