

ornl

**OAK RIDGE
NATIONAL
LABORATORY**

LOCKHEED MARTIN 

RECEIVED
SEP 03 1997
OSTI

Demonstration of Volumetric Analysis Using the Topographical Mapping System at Hanford

**Gary A. Armstrong
Barry L. Burks
Betty A Carteret*
Allan F. Pardini*
Todd J. Samuel***

***Pacific Northwest National Laboratory**

MASTER 
DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

**MANAGED AND OPERATED BY
LOCKHEED MARTIN ENERGY RESEARCH CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY**

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615) 576-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible electronic image products. Images are produced from the best available original document.

Robotics and Process Systems Division

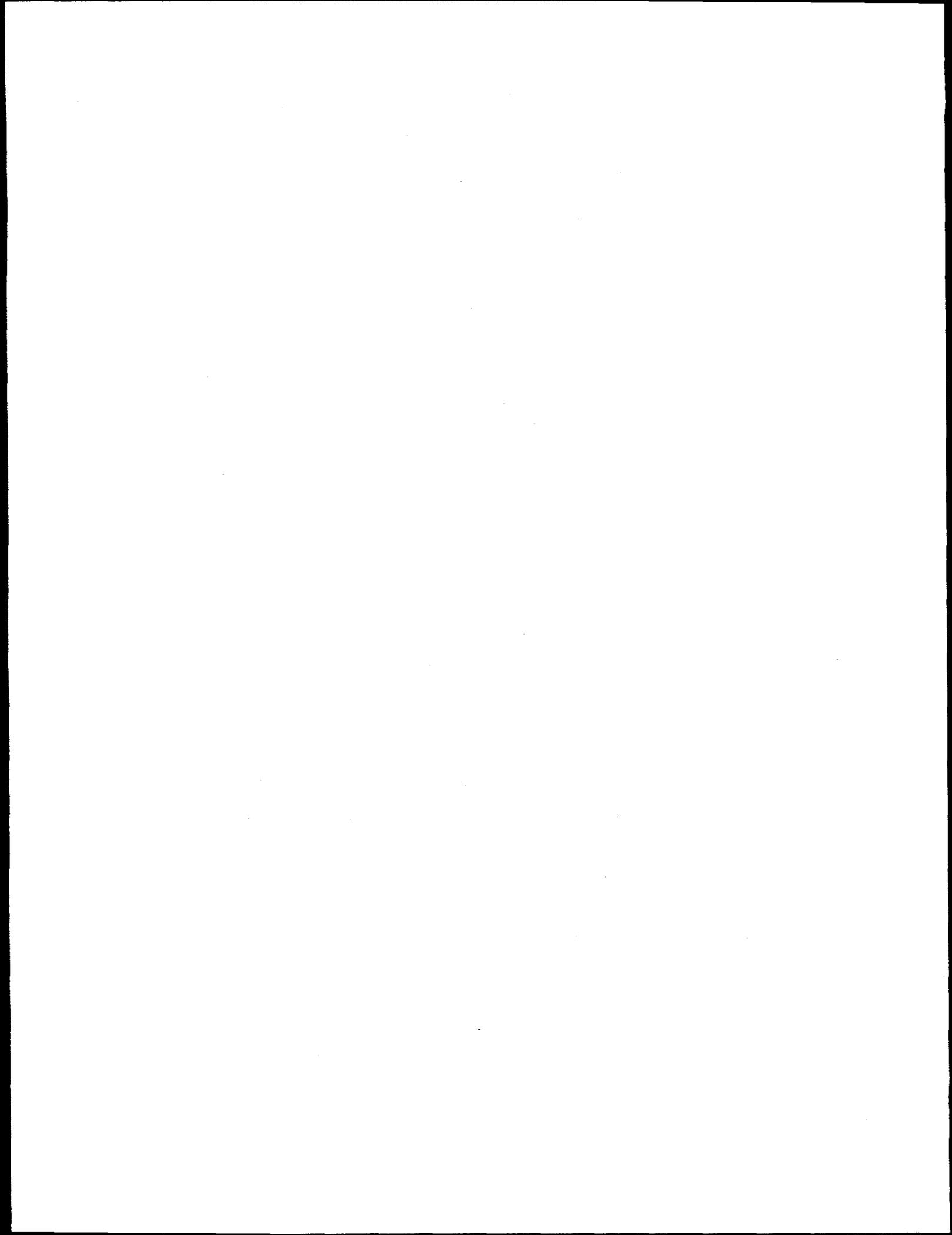
**DEMONSTRATION OF VOLUMETRIC ANALYSIS USING THE
TOPOGRAPHICAL MAPPING SYSTEM AT HANFORD**

Gary A. Armstrong
Barry L. Burks
Betty A. Carteret*
Allan F. Pardini*
Todd J. Samuel*

* Pacific Northwest National Laboratory

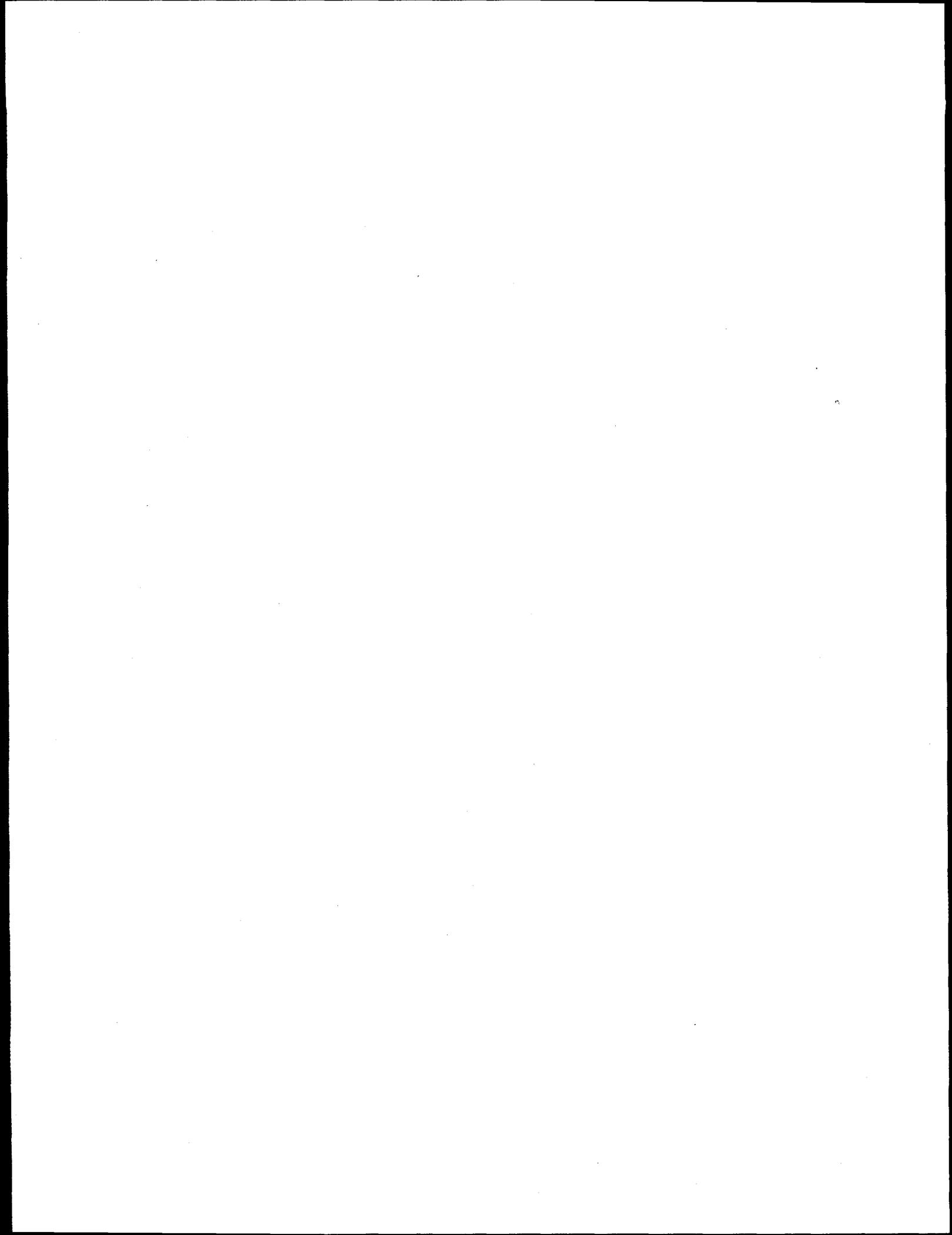
Date Published—July 1997

Prepared by
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831-6285
Managed by
LOCKHEED MARTIN ENERGY RESEARCH CORP.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-96OR22464



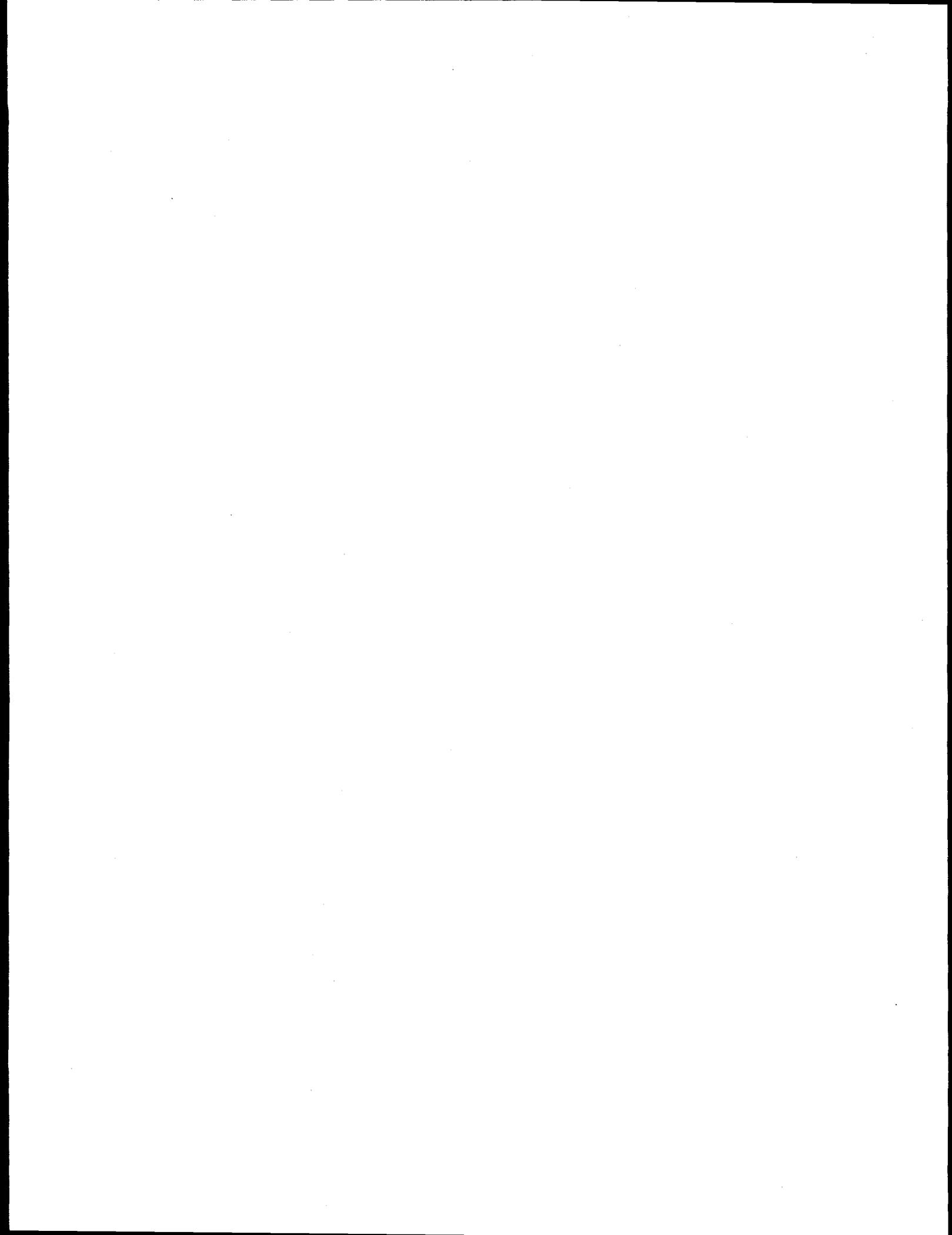
CONTENTS

LIST OF FIGURES	v
ACRONYMS	vii
ABSTRACT	ix
1. INTRODUCTION.....	1
1.1 HANFORD USTs.....	1
2. TOPOGRAPHICAL MAPPING SYSTEM.....	3
2.1 SYSTEM REQUIREMENTS	3
2.2 SYSTEM DESCRIPTION	4
2.3 TMS CALIBRATION AND CHARACTERIZATION	8
3. VOLUMETRIC ANALYSIS	11
3.1 GENERAL	11
3.2 SOFTWARE	11
3.3 RESULTS.....	12
4. FUTURE WORK.....	15
5. FUTURE PLANS.....	17
6. REFERENCES.....	19
7. SELECTED BIBLIOGRAPHY	21
APPENDIX A: TMS SURFACE MAPS	23
APPENDIX B: TMS DIAGRAMS AND PHOTOGRAPHS	37
APPENDIX C: TYPICAL UST DIMENSIONS.....	45
APPENDIX D: VOLUMETRIC SOURCE CODE	51



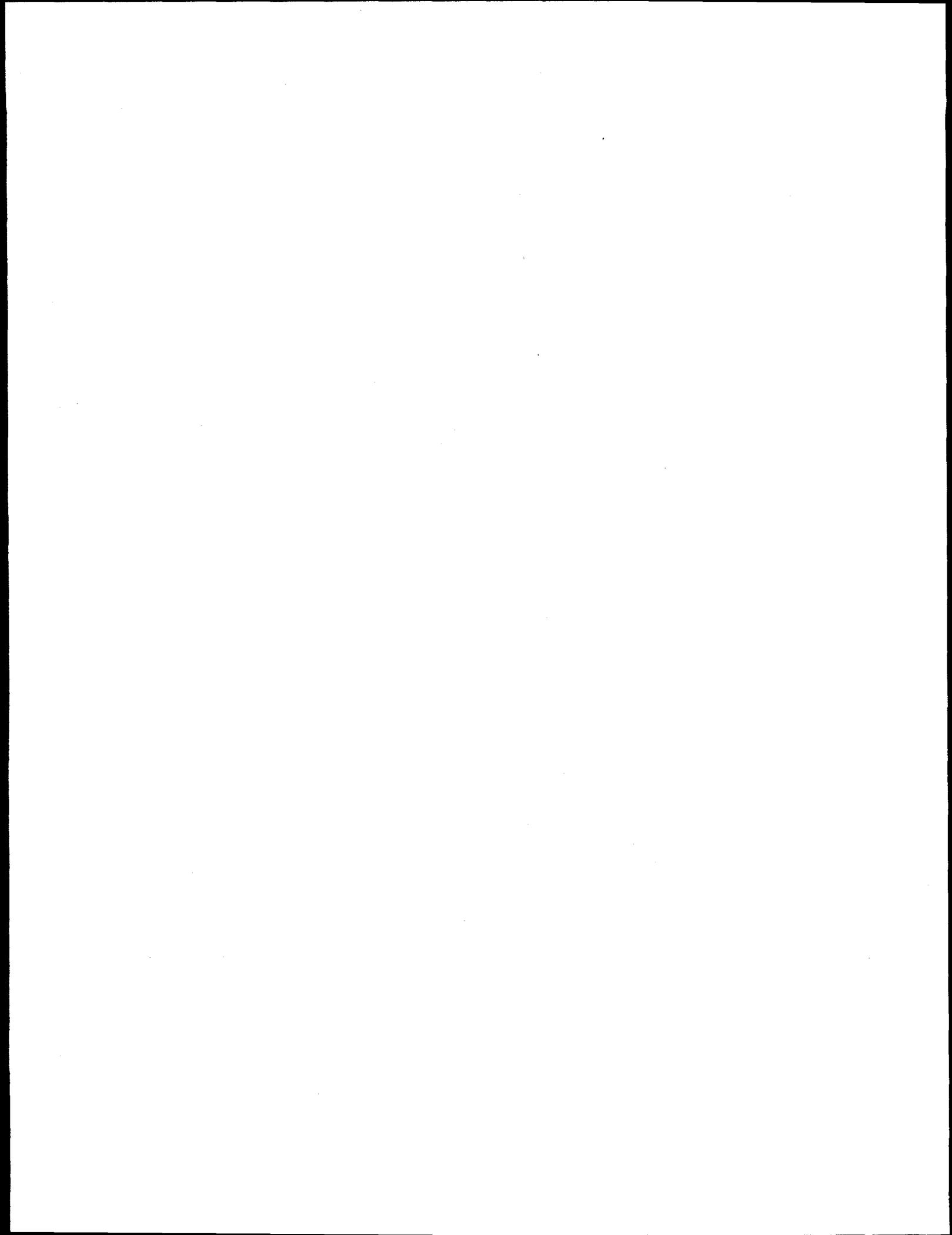
LIST OF FIGURES

Figure	Page
1 Time-lapse photograph of the structured-light mapping process	5
2 TMS block diagram	6
3 TMS sensor head deployed in a UST	7
4 Relative accuracy of the TMS	9
5 TMS deployed in the FMEF cold test cell	13
6 Sandbox with mounds 1 through 4 used for volumetric analysis.....	14
A.1 Table top surface map with sand mound.....	25
A.2 Table top difference map.....	27
A.3 Left sandbox surface map with mounds 1, 2, and 3	29
A.4 Left sandbox difference maps of mounds 1, 2, and 3.....	31
A.5 Right sandbox surface map with mound 4	33
A.6 Right sandbox difference map of mound 4	35
B.1 Block diagram of the TMS	39
B.2 TMS cable diagram	40
B.3 TMS laser/camera module.....	41
B.4 TMS pan motor adapter module.....	42
B.5 TMS deployed on the bridge of the Robotics and Process Systems Division's Technology Test Facility.....	43
B.6 TMS control station.....	43
B.7 Surface map of MTI's Optics Lab, Albany, New York, displayed with ICERVS.....	44
C.1 Typical configuration of a 75-ft type C, B, BX, T, and U single-shell tank	47
C.2 Typical configuration of a 75-ft type SX, A, and AX single-shell tank.....	48
C.3 Typical configuration of a 75-ft type TX, BY, S, and TY single-shell tank.....	48
C.4 Schematic showing the access pipe from the surface to the tank shell.....	49



ACRONYMS AND ABBREVIATIONS

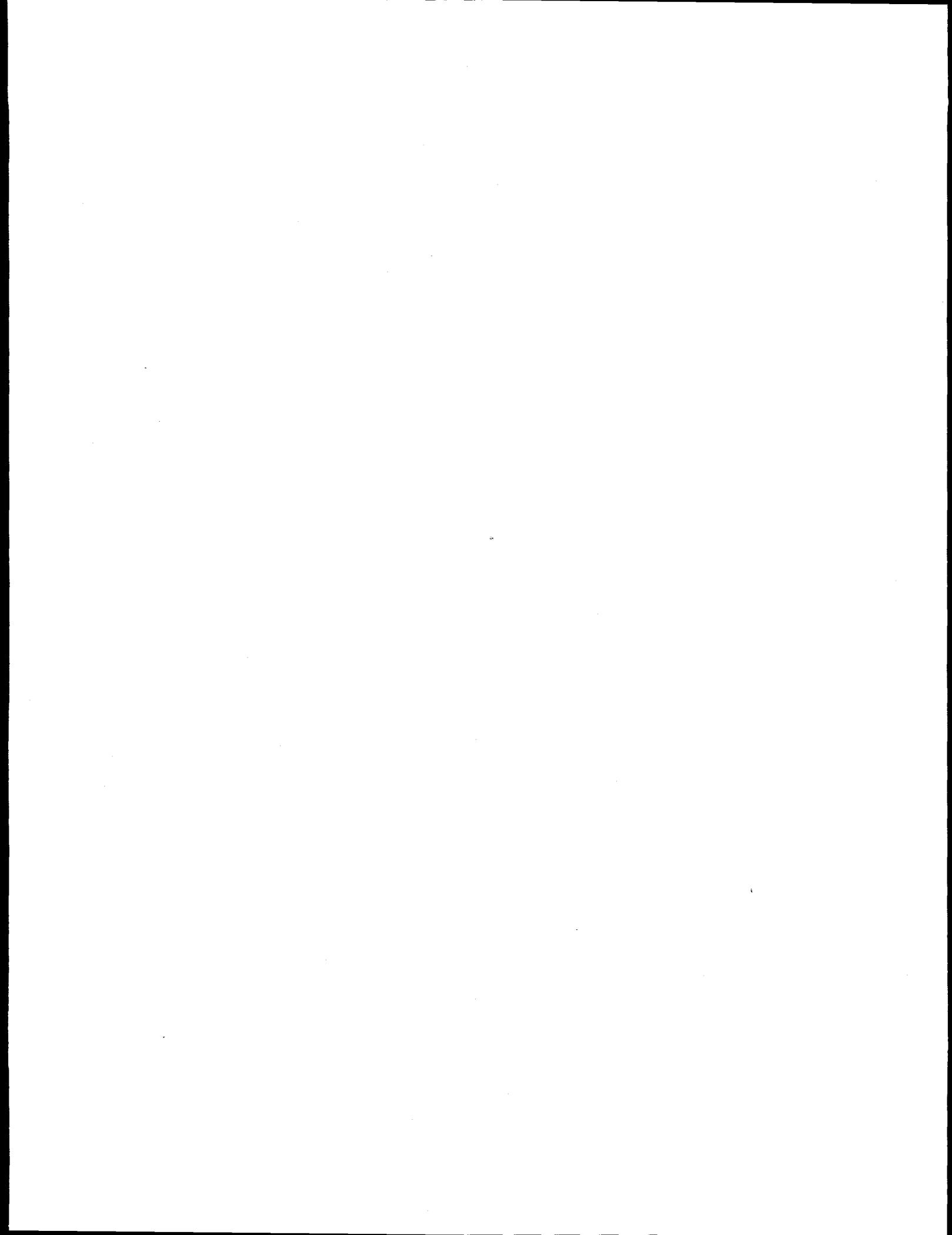
2-D	two-dimensional
3-D	three-dimensional
CCD	charge-coupled device
CRADA	cooperative research and development agreement
DOE	U.S. Department of Energy
EEB	environmental enclosure box
ESS	environmental sensor section
FETC	Federal Energy Technology Center
FMEF	Fuel Materials and Examination Facility
HMI	human-machine interface
ICERVS	Interactive Computer-Enhanced Remote-Viewing System
MTI	Mechanical Technology, Inc.
ORNL	Oak Ridge National Laboratory
PNNL	Pacific Northwest National Laboratory
RMS	root mean square
SST	single-shell tank
STF	South Tank Farm
TMS	Topographical Mapping System
UST	underground storage tank
WHC	Westinghouse Hanford Company



ABSTRACT

During the spring of 1997, the Topographical Mapping System (TMS) for hazardous and radiological environments was used to perform volumetric measurements of simulated waste in the cold test cell in the Fuel Materials and Examination Facility at the Hanford site. The TMS was used to measure the volume of five simulated waste mounds. Custom software designed by Oak Ridge National Laboratory was used to calculate the volume of waste from the surface maps supplied by the TMS. The results of the measurements were analyzed using the Interactive Computer-Enhanced Remote-Viewing System (ICERVS) and were documented.

Development of the TMS and ICERVS was initiated by the U.S. Department of Energy (DOE) for the purpose of characterization and remediation of underground storage tanks (USTs) at DOE sites across the country. DOE required a three-dimensional TMS suitable for use in hazardous and radiological environments. The intended application is the mapping of the interior of USTs as part of DOE's waste characterization and remediation efforts to obtain baseline data on the content of storage tank interiors as well as on changes in the tank contents and levels brought about by waste remediation steps. Initially targeted for deployment at the Hanford site, the TMS was designed to be a self-contained, compact, and reconfigurable system that is capable of providing rapid, variable-resolution mapping information in poorly characterized workspaces with a minimum of operator intervention.



1. INTRODUCTION

This report focuses on the use of the Topographical Mapping System (TMS) for hazardous and radiological environments and the Interactive Computer-Enhanced Remote-Viewing System (ICERVS) to perform volumetric measurements of simulated waste in the cold test cell in the Fuel Materials and Examination Facility (FMEF) at the Hanford site in Washington State.

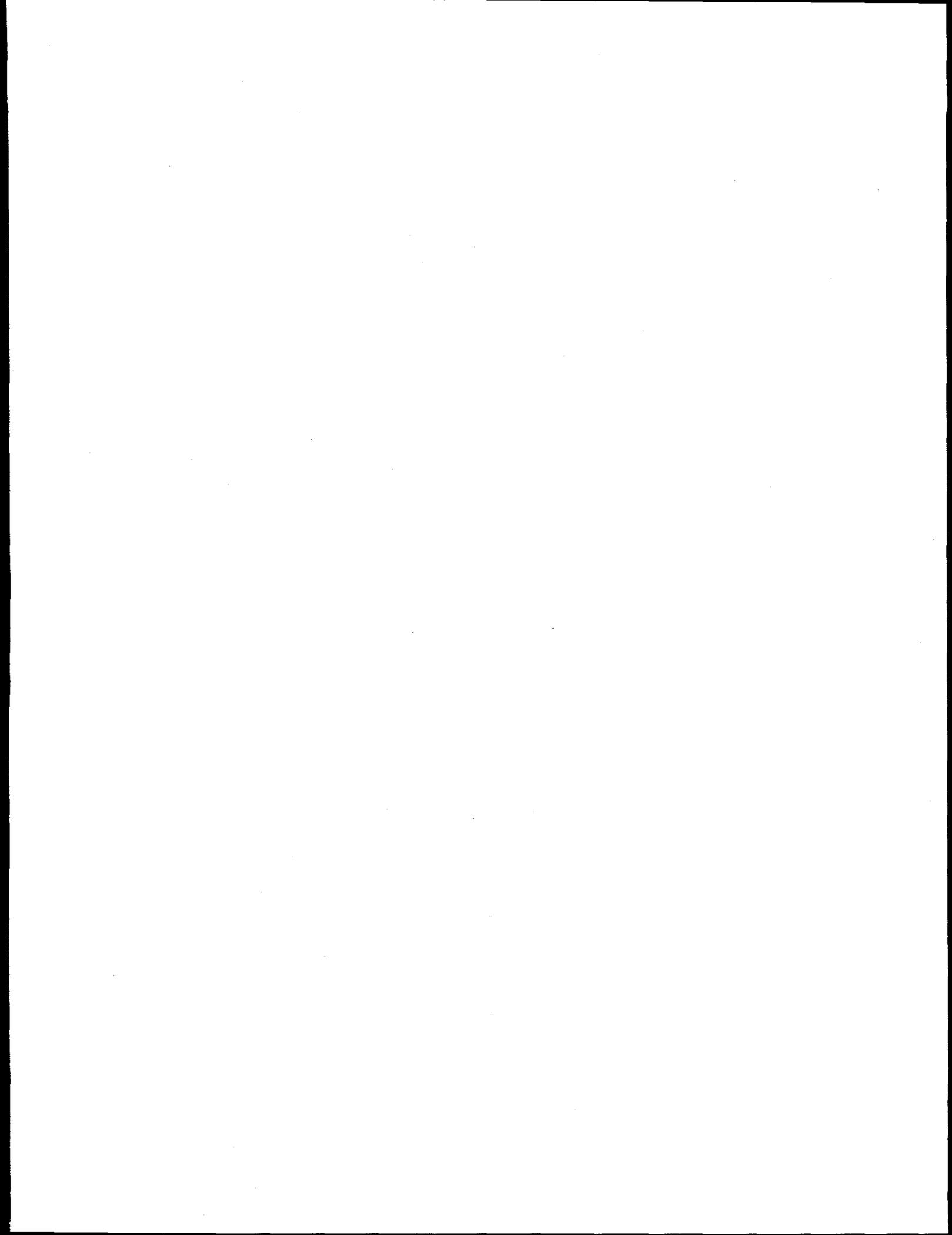
The TMS was developed by the U.S. Department of Energy (DOE) Tanks Focus Area under the sponsorship of the Office of Science and Technology (EM-50). The DOE Robotics Technology Development Program funded the Oak Ridge National Laboratory (ORNL)-developed surface mapping system (deployed in the K-65 storage tanks at Fernald in 1991) and the prototype surface mapping system (demonstrated at the Hanford site in 1993), which was developed under a cooperative research and development agreement (CRADA). The ICERVS tool was developed by the Federal Energy Technology Center (FETC).

1.1 HANFORD USTs

Underground Storage Tanks (USTs) at DOE sites, such as the Hanford¹ site in southeastern Washington State, contain hazardous, radioactive waste generated during defense material production over the past 50 years. A number of the tanks have been used past their intended design life and are deteriorating, contaminating the surrounding environment. Stabilization and remediation of these tanks is a high priority for the DOE Environmental Restoration Program. The TMS will gather vital data needed to respond to ongoing questions about the safe storage of waste materials and to quickly investigate tank events, such as leaks, that raise safety concerns.

There are 149 single-shell tanks (SSTs) at the Hanford site. The SSTs range in size from 208,000 to 3,800,000 L (55,000 to 1,000,000 gal). The tanks are cylindrical, constructed with reinforced concrete, and lined with carbon steel. The 208,000-L tanks (there are 16 total) have flat tops and are 6 m (20 ft) in diameter. The larger tanks have domed tops and are 23 m (75 ft) in diameter. Tank heights vary depending on capacity. There is considerable variation in the number, location, and type of openings in the tops of the tanks, with the majority having at least one 107-cm-diam (42-in.-diam) opening in the center and one or more 10.16-cm (4-in.) opening around the periphery. The top of a typical tank is 2.5 m (8 ft) below grade.

The original waste consisted of liquids (strong acids) from the plutonium separation process. The acids were neutralized before they were put into the tanks. The neutralization process caused a complex mixture of solids to precipitate and form a layer of sludge on the bottom of the tanks. To reduce the volume of the liquids, as well as to remove radioactive isotopes of cesium and strontium, a waste reduction process was initiated in the 1960s. This program significantly reduced the amount of water (using evaporation) and reduced the concentration of cesium and strontium. The result was a concentrated salt slurry, which was returned to the USTs. The salt slurry is now salt cake. Although not fully characterized, it is estimated that radiation levels near the surface of the salt cake in a typical tank are less than 100 1 Gy/h (100 rad/h). Currently, the USTs at Hanford contain residual liquids and sludges from past radiochemical separation processes.



2. TOPOGRAPHICAL MAPPING SYSTEM

In 1991 ORNL developed and deployed a structured-light-based surface mapping system in the K-65 tanks at the Fernald site.² The structured-light surface mapping system was used to determine the waste surface topography before and after the addition of a clay cap. The data were used to ensure that the clay cap applied over the waste was a minimum of 30.48 cm (12 in.) deep at all locations per U.S. Environmental Protection Agency requirements. In 1993 ORNL and Mechanical Technology, Inc. (MTI), Albany, New York, initiated a CRADA for the development of a structured-light surface mapping system for deployment in the USTs at the Hanford site. The successful CRADA demonstration at Hanford in June 1994 proved that a structured-light surface mapping system could be built to penetrate a 10.16-cm (4-in.) clear aperture and map the insides of a UST to ranges of 13.7 m (45 ft) with an accuracy of 0.64 cm (0.25 in.). Based on the results of the deployment at Fernald and the CRADA, a request for proposals was generated in February 1994 to develop a surface mapping system that could withstand the radiological and hazardous environments at the Hanford site. The contract was placed with MTI in May 1994. The system was delivered to ORNL in June 1996 for acceptance testing, which was completed in February 1997. The TMS was deployed in tanks W5 and W6 of the South Tank Farm (STF) at ORNL in February 1997.³ After deployment at ORNL, the system was delivered to the Hanford site and was used to demonstrate volumetric measurement of waste in the FMEF cold test facility in March 1997.

The development of the TMS was initiated by DOE for the purpose of characterization and remediation of USTs at DOE sites across the country. DOE required a three-dimensional (3-D), surface mapper suitable for use in hazardous and radiological environments. The intended application is the mapping of the interior of USTs as part of DOE's waste characterization and remediation efforts to obtain baseline data on the content of the storage tank interiors as well as on changes in the tank contents and levels brought about by waste remediation steps. Because TMS was initially targeted for deployment in USTs at the Hanford site, applicable definitions of USTs can be found in the Westinghouse Hanford Company (WHC) documents WHC-EP-0352, "Single Shell Tank Waste Retrieval Study," and WHC-SD-RE-TI-053, "Riser Configuration Document for Single Shell Tanks."^{4,5} "The Topographical Mapping System for Hazardous and Radiological Environments Statement of Work,"⁶ defines the performance specifications of the TMS and the environmental conditions under which the TMS must operate. The TMS was designed to be a self-contained, compact, and reconfigurable system that is capable of providing rapid, variable-resolution mapping information in poorly characterized workspaces with a minimum of operator intervention.

2.1 SYSTEM REQUIREMENTS

The primary purpose of the TMS is to generate reliable, registered, and accurate 3-D maps of the internal surfaces of a UST. In addition to the walls, dome, and waste, these tanks contain salt pumps, air circulator risers, thermocouple trees, and objects that have fallen or been placed in the tanks. One use for this mapping system is in creating and maintaining a current 3-D map of the tank interior as input to a robotic "world model" that is used to test remediation strategies or plan robot trajectories. Another use is tracking the movement of the waste surface as it responds to expanding bubbles of trapped gas. A third use is to perform a volumetric analysis of the amount of waste removed from the tanks during remediation by mapping the waste before and after remediation activities. A fourth use of the TMS is in determining how much waste is left in the tank. The fourth application is dependent on accurate drawings of the tank or a method by which an accurate description of the tank structure can be constructed.

Performance requirements are based on the "Functions and Requirements for the Light-Duty Utility Arm Integrated System"⁷ document from WHC and Pacific Northwest National Laboratory (PNNL) along with insights and lessons learned at ORNL through previous surface mapping projects. The primary requirements are as follows:

- Accuracy requirements may vary considerably. For example, to track the movement of the waste surface, it may be necessary to measure changes as small as 2.54 mm (0.10 in.). For collision avoidance, measurement errors of 101.6 mm (4.0 in.) are acceptable. The TMS has been specified to provide an accuracy of ± 6.35 mm (± 0.25 in.) over a range of up to 13.7 m (45 ft).
- Mapping data densities should be at least one point per 150- by 150-mm (6- by 6-in.) region, covering up to 95% of the surfaces in the tank. The time required for mapping cannot exceed 2 hours at this data density, although more time would be allowed for mapping at higher densities. The highest density that the TMS is required to provide is one point in every 25.4- by 25.4-mm (1.0- by 1.0-in.) region of surface (the present system can provide one point in every 2.54- by 2.54-mm (0.10- by 0.10-in.)).
- The TMS has been specified to operate in a continuous flux of 5 Gy/h (500 rad/h) and in an intermittent peak flux of 10 Gy/h (1000 rad/h) up to a total absorbed dose of 1E4 Gy (1E6 rad) without failure caused by irradiation (this requirement can also be met by periodic replacement).
- The TMS has been specified to be deployed through a 88.9-mm (3.5-in.) clear aperture to allow deployment through the 101.6-mm (4-in.) risers at the Hanford site but can also be deployed through the 304.8-mm (12-in.) or larger risers.
- The TMS has been specified to be Class 1, Division 1, Group B, hazardous environment compliant to permit the use of the TMS in tanks that contain volatile gases.
- A temperature range of 10 to 50°C (50 to 122 °F) with a noncondensing relative humidity of 100% has been specified to allow the TMS to operate in the varying environments that may be found in the tanks at the Hanford site.

2.2 SYSTEM DESCRIPTION

The TMS is a distributed architecture computer-based system that also collects temperature and radiation flux measurements and that has a single-point laser range finder. The topographical mapping sensor uses structured light. Structured light is a triangulation-based range measurement technique. The structured-light measurement technique projects a laser plane onto the surface to be mapped. The resulting intersection of the laser plane and the surface produces a contour line annotating the shape of the surface. A camera is used to image the resulting laser plane's contour line. Figure 1 is a time-lapsed photograph of the TMS scanning the laser over a simulated waste surface of sand and salt cake. The charge-coupled device (CCD) camera has a vector assigned to each pixel in the CCD array. Every point that is illuminated by the laser line reflection is passed to the analytical routines for processing. The analytical routines solve for the intersection of the fixed vector assigned to the pixel in the camera with the equation of the plane of the laser (each intersection of a vector and the laser plane produces an $\langle XYZ \rangle$ point in space). In summary, by combining the range measurement with the kinematics of the sensor head, the TMS is able to determine the $\langle XYZ \rangle$ description of points located on the surfaces of the interior of the USTs with respect to a world-coordinated frame typically located at the bottom center of the tank.



Fig. 1. Time-lapse photograph of the structured-light mapping process. This photograph illustrates the contour line formed when the laser plane intersects the surface of the simulated waste. The simulated waste includes sand, simulated salt cake (white rock), and two vertical pipes.

The TMS has four major components:

1. The sensor head contains the optical metrology sensors that penetrate the vapor space of the tank and provide the topographical map of the interior surfaces.
2. The environmental enclosure box (EEB) contains all the support electronics that require proximity to the sensor head (e.g., the frame grabber and motor controllers and the local computer that runs them).
3. The human-machine interface (HMI) is located in the control trailer approximately 274 m (900 ft) away and is used for supervisory control, limited data visualization, and data archiving. The HMI is a UNIX-based scientific and engineering workstation that provides the graphical operator interface and supports the various command, control, and communication functions required for proper system function.
4. The plug gauge is used to test the clear aperture of the riser before deployment of the sensor head. The plug gauge also contains the environmental sensor section (ESS), which provides measurements of temperature, radiation, and range that are used to deploy the sensor head. (The ESS can also be

attached to the distal end of the sensor head, or the sensor head can be deployed with a dummy ESS module.)

The system block diagram is illustrated in Fig. 2, and the sensor head is shown deployed in a UST in Fig. 3.

Other systems that are needed to deploy the TMS, which were not included as part of the TMS contract, include the purge gas supply and withdrawal system, the structure used to position and hold the TMS sensor head over the riser (such as a trailer and strong-back), containment systems used to contain the riser openings, containment storage structures, and a 3-D visualization system used to analyze the data. Currently, the ICERVS is used for 3-D visualization and data analysis.⁸ The ICERVS was developed by MTI for use in DOE characterization and remediation efforts under contract to FETC. The ICERVS tool allows for display and analysis of the unusually large data sets generated by mapping USTs that can be as large as 22.86 m (75 ft) in diameter with 7.92-m (26-ft) walls. Mapping the walls, floor, and dome of a typical 22.86-m-diam (75-ft-diam) empty tank would generate 3 million data points (or a 180-MB file). These additional systems needed for deployment (except for the ICERVS) have to be supplied on site by the user because they do not currently exist as part of the TMS.

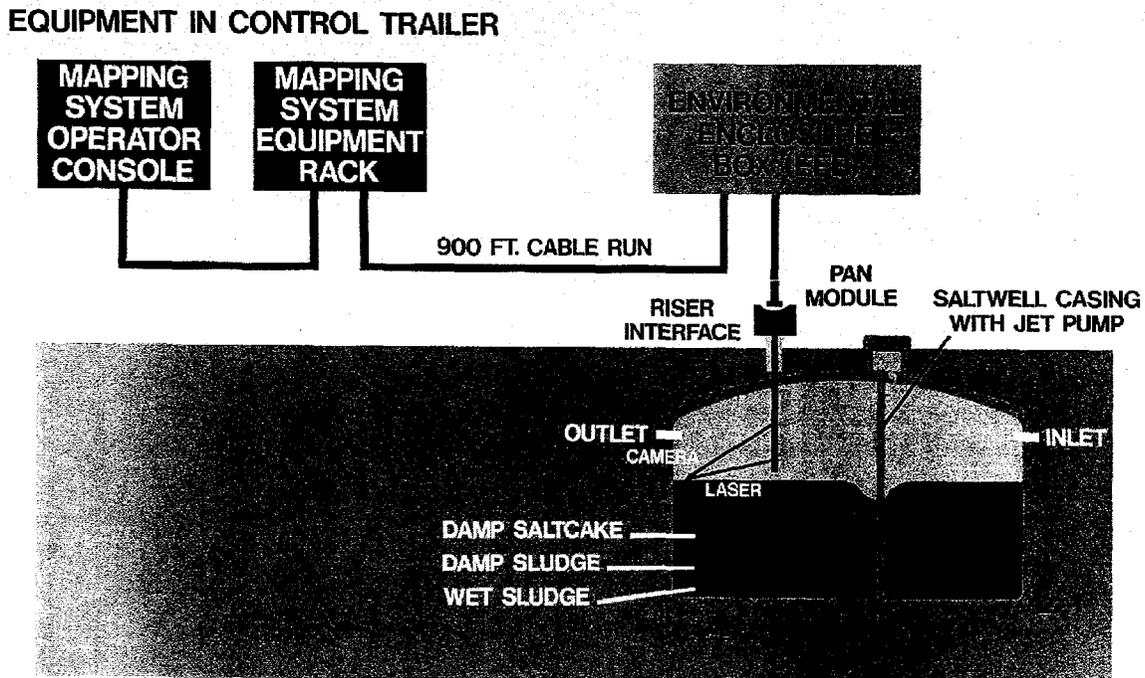


Fig. 2. TMS block diagram.

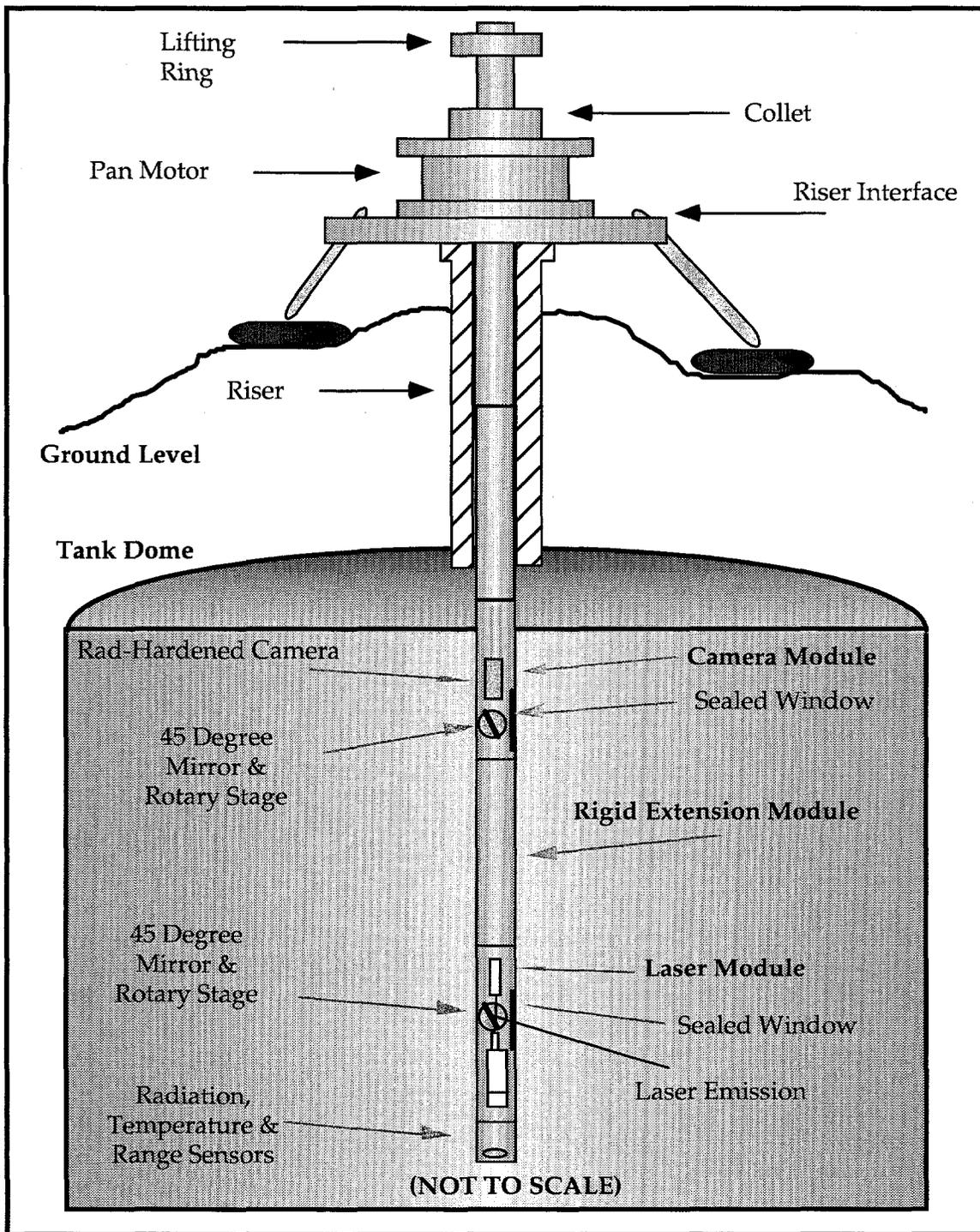


Fig. 3. TMS sensor head deployed in a UST.

2.3 TMS CALIBRATION AND CHARACTERIZATION

Before the tank wall inspection performed in the STF USTs at ORNL, the TMS was calibrated and characterized by MTI and ORNL staff. The TMS was calibrated and characterized through the use of the high bay and basement in the Robotics and Process Systems Division's facility at ORNL. The TMS sensor head was deployed through a 10.16-cm (4-in.) hole bored in the floor between the high bay and the basement. The high bay provided access to the pan motor and the EEB and allowed for overhead crane deployment of the TMS. The testing area in the basement is approximately 18 m (60 ft) long and 9 m (30 ft) wide. The floor of the basement to the lowered floor of the platform in the high bay is 6.1 m (20 ft), which puts the TMS camera at 5.33 m (17 ft, 6 in.) above floor level. This provides for a maximum range of 10.57 m (34 ft, 8 in.) from a diagonal that originates at the camera at the proximal end to the bottom of the wall 9.14 m (30 ft) away at the distal end. The minimum range was 5.76 m (18 ft, 11 in.) because of the 22-degree, half-angle occlusion area directly beneath the TMS sensor head. This permits angles of incidence that range from 22 to 60 degrees.

The floor of the basement was marked using a metal measuring tape that was positioned for alignment with the Y axis of the TMS coordinate system in its home position. The measuring tape was aligned using a theodolite, which was centered on the TMS sensor head. The theodolite ensured that the measuring tape was positioned directly along the Y axis of the TMS. This permitted measurements made in the measuring tape coordinate frame to be mapped into the TMS coordinate frame for comparison purposes. To further facilitate the verification and testing of the TMS's ability to accurately measure points in the basement, a Pentax PTS-V Total Station⁹ was used to accurately survey points. After registering the Total Station coordinate space and the TMS coordinate space to a known reference space, the surveyed points were mapped to TMS coordinate space for direct comparison with the TMS sensor.

The first part of the calibration involved the alignment of the laser and camera modules on their respective rotary stages. The second part of the calibration determined the fixed vectors that are assigned to each camera pixel and the kinematics of the laser and camera rotary stages. The camera vectors are determined by the two-plane camera calibration method.^{10, 11}

The final part of the TMS calibration was to calibrate the five coordinate frames between the three joints and the various modules used to configure the TMS sensor head. The first deployment of the TMS was scheduled for the USTs in the STF at ORNL. As a result, the six calibration targets were placed in positions to optimize the TMS's measurement accuracy for measuring degradations in the UST walls of tanks W5 and W6 in the STF. Tanks W5 and W6 are 15.24 m (50 ft) in diameter with 3.66-m (12-ft) walls and are capped with domes that crest 1.8 m (6 ft) above the walls. The central riser extends 2.13 m (7 ft) up past the dome that is then bermed with dirt. The height from the top of the central riser to the bottom of the tank is 7.62 m (25 ft). For the ORNL deployment, only the walls of the tank would be mapped. (The first deployment at ORNL was to measure cracks and spalling concrete as well as signs of structural instabilities in the UST walls, so the TMS was optimized for these measurements.)

Results of the characterization of the TMS are shown in Fig. 4. The relative root mean square (RMS) error varied from 10.16 mm (0.40 in.) at 3.66 m (12 ft) to 6.35 mm (0.25 in.) at the optimized 7.62 m (25 ft). The system error was driven primarily by the lateral error. The axial or range error was very low, ranging from 0.43 mm (0.02 in.) at 5.79 m (19 ft) to 1.8 mm (0.07 in.) at 8.84 m (29 ft). Because the cubic box that was used to make the measurement was placed on the center of the field of view (which was necessary to keep the entire box within the 10-degree fan angle), the surface map from which the range measurements were derived was always created down the center of the field of view. Because the misaligned laser had the biggest visible effect in skewing the resulting surface map, the largest errors would then be driven by the lateral measurements, which is the case here.

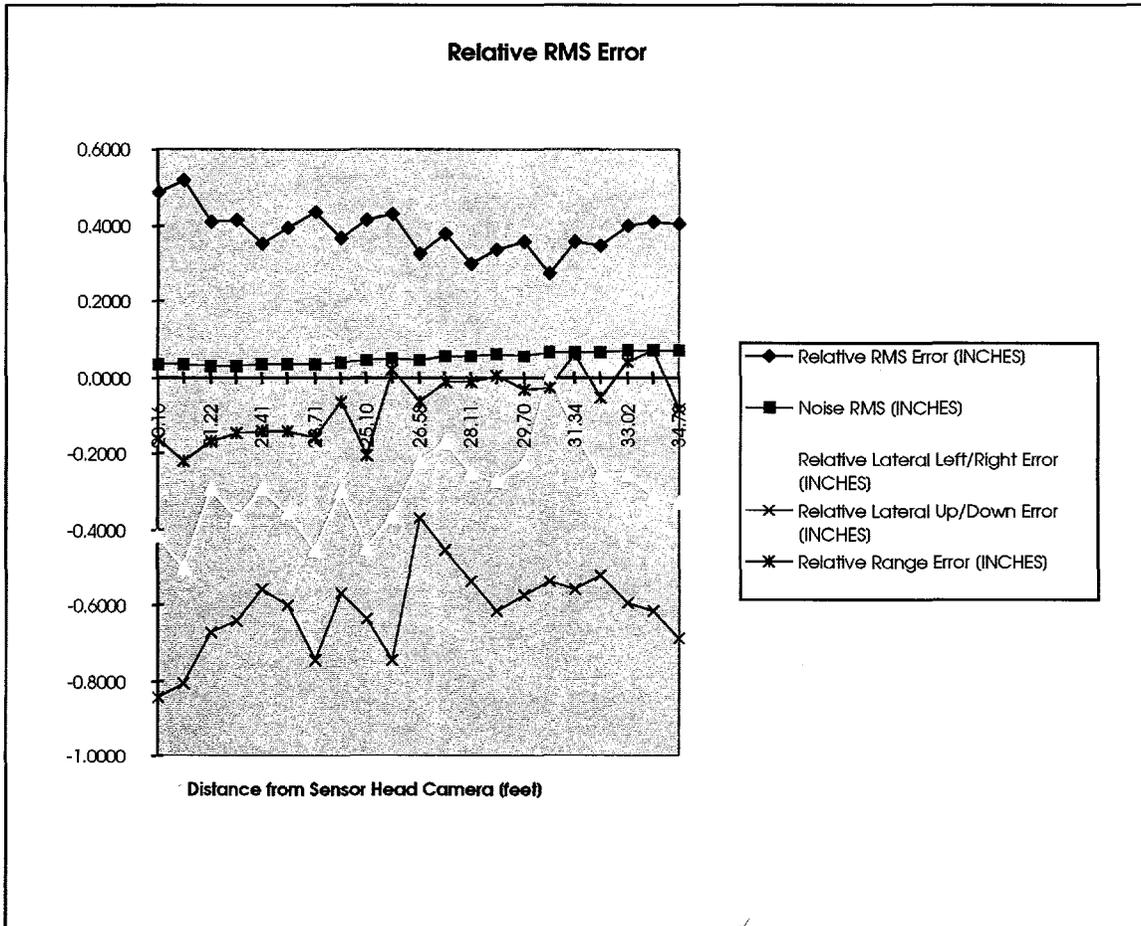
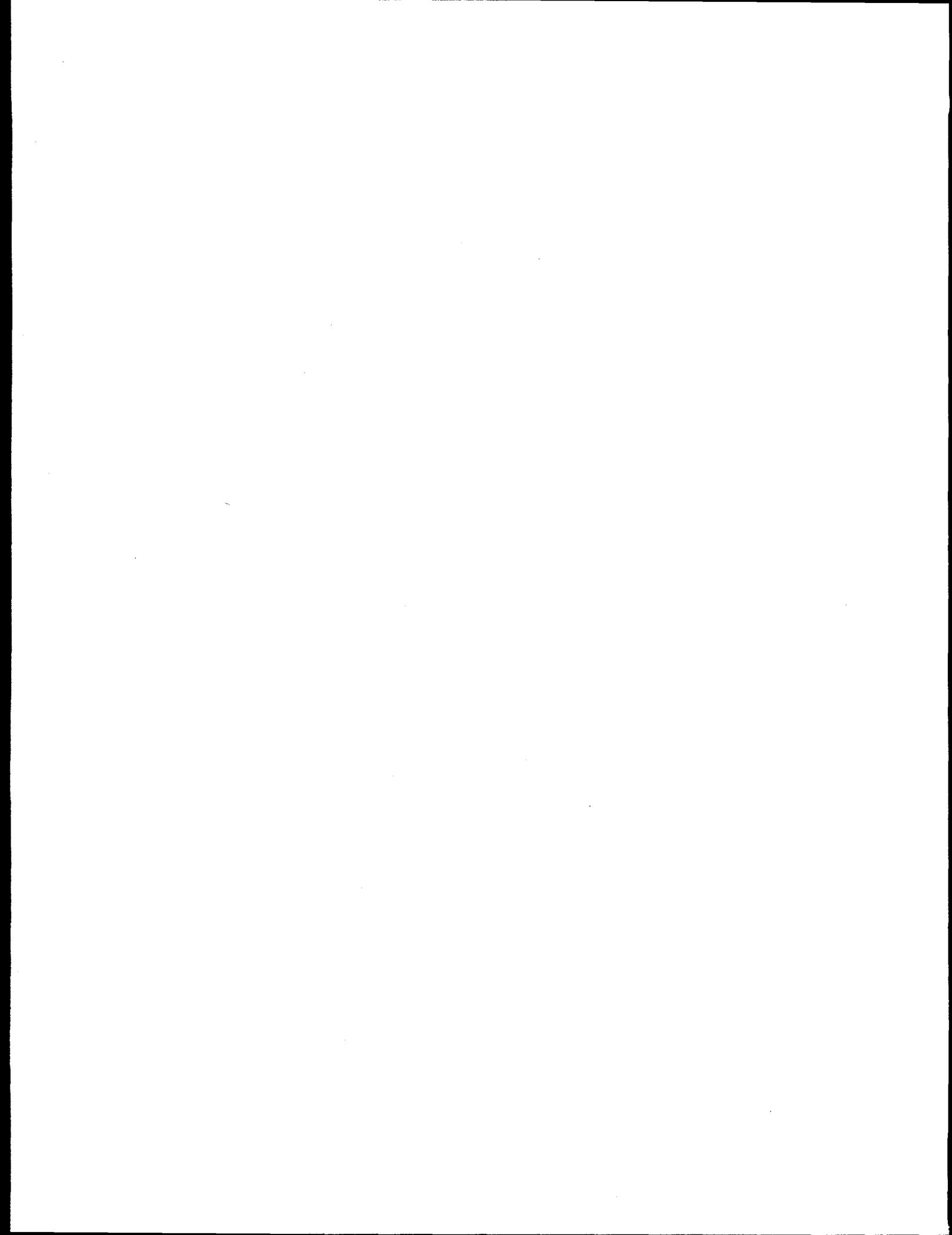


Fig. 4. Relative accuracy of the TMS.



3. VOLUMETRIC ANALYSIS

3.1 GENERAL

Volumetric analysis can serve two purposes in a UST. The first is to determine the amount of waste removed from the UST for accountability purposes. This method requires that the waste surface be mapped before and after the removal of waste from the UST. The second use of volumetric analysis is to measure the amount of waste left in the tank. This requires that the bottom of the UST be known either by drawings or by other measurement techniques. Once the bottom of the UST is known, a surface map of the remaining waste can then be registered to a known coordinate frame and the two surfaces can be subtracted to determine the waste's volume.

The volumetric analysis performed at PNNL during March 1997 used the before-and-after surface map technique. In each case the surface was mapped before and after the application of simulated waste material (i.e., sand). The sand was poured in piles that varied from approximately 2.2 to 30.8 L (2 to 28 quarts).

Appendix A contains color photographs of the surface maps used to calculate the volumes in this report. Appendix B contains diagrams and photographs of the TMS. Appendix C contains illustrations of typical tank dimensions. Appendix D contains the custom ORNL source code used to calculate the volumes from the surface data.

3.2 SOFTWARE

The volumetric measurement of the surface maps data was performed by customized software developed by ORNL. The software loaded the before-and-after surface maps into two-dimensional (2-D) arrays, performed preprocessing of the data, and then subtracted the two maps. The resulting "difference" map was used to determine the volume of the sand by multiplying the height of each point by the area that each point represented; for example, for a 0.635-cm (0.25-in.) mapping resolution, the area would be 0.403 cm^2 (0.0625 in.^2).

Because the TMS sensor head does not collect data in an arrayed format and there is therefore no implied connectivity between collected points, the data were placed into the 2-D array cells or bins by X and Y values. As a result, some of the bins in the array were left vacant while other bins accumulated multiple data values. The accumulated data were averaged to determine the resulting value for each bin. In addition, occlusions in the scene left small areas in the array with no data. The preprocessing algorithms filled in all the empty and occluded regions by two methods: (1) averaging with a convolution mask and (2) linear interpolation. The averaging process was able to fill in many of the areas where only one data point was missing in addition to acting as a low-pass filter by smoothing out the surface map. The averaging algorithm swept a 3-by-3 convolution mask in raster format over the entire array. At each point, the eight nearest neighbors along with the value of the point itself (if there was one) were averaged together.

A linear interpolation algorithm was used to fill in the larger empty areas caused by occlusion. The algorithm would look left and right as well as up and down in the array rows and columns at each occluded point until it found occupied bins. The resulting points were used to linearly interpolate an estimated value for the occluded point.

Once all the points in the two arrays were filled, the two arrays could be subtracted and the volume then determined from the difference array.

3.3 RESULTS

The difference maps for mapping the sand mounds on the table top and the sand pit are shown in Appendix A. The accuracy of the results ranges from 75 to 93% of the targeted results. Further study is required to examine all the possible sources of error. Part of the inaccuracy comes from the imprecise nature in which the sand was measured. The sand was measured with a 1.1-L (1-quart) plastic container. The container was filled by hand to the 1.1-L (1-quart) mark on the side that was measured visually. No measurements were made to determine whether the sand was compressing during measurement or application. After the sand contained in the plastic quart container was measured with a machined container, it was estimated that the plastic container held 909 cm^3 (55.47 in.^3) of sand. No measurements were made on the repeatability of this measurement. The second known source of error comes from the TMS itself. As demonstrated from the relative accuracy data in Fig. 4, the average relative accuracy of the TMS is 1 cm (0.4 in.) at 9.1 m (30 ft) from the camera. For a mound that peaks at 5.6 cm (2.2 in.) (table-top mound) the error can be roughly approximated at 17.5%. Note that the table-top mound and mound 4 of the sand pit produced results that were 93 and 86% the targeted results, respectively; while mounds 2 and 3 produced results of 75 and 77%, respectively. Because mounds 2 and 3 were much smaller than the other mounds, the TMS error of 1 cm (0.4 in.) had a larger effect on the results. Mound 1 was comparable in size to the table-top mound and mound 4 but had an 81% accuracy result. This is thought to be partially caused by the volumetric software's linear interpolation algorithm. Mound 1 had the largest occlusion area of all the mounds that were mapped. The linear interpolation does not correctly model the curved surface of the mound and as a result throws away some of the volume of the mound. A second-order interpolation would have produced much better results.

Figure 5 shows the TMS sensor head deployed in the FMEF test cell. Figure 6 shows the sandbox with mounds 1 through 4.

The results from the volumetric analysis are summarized in Table 1. The data show the maximum height of each sand mound as well as the approximate amount of sand used to build the mound ("Container Measurement").

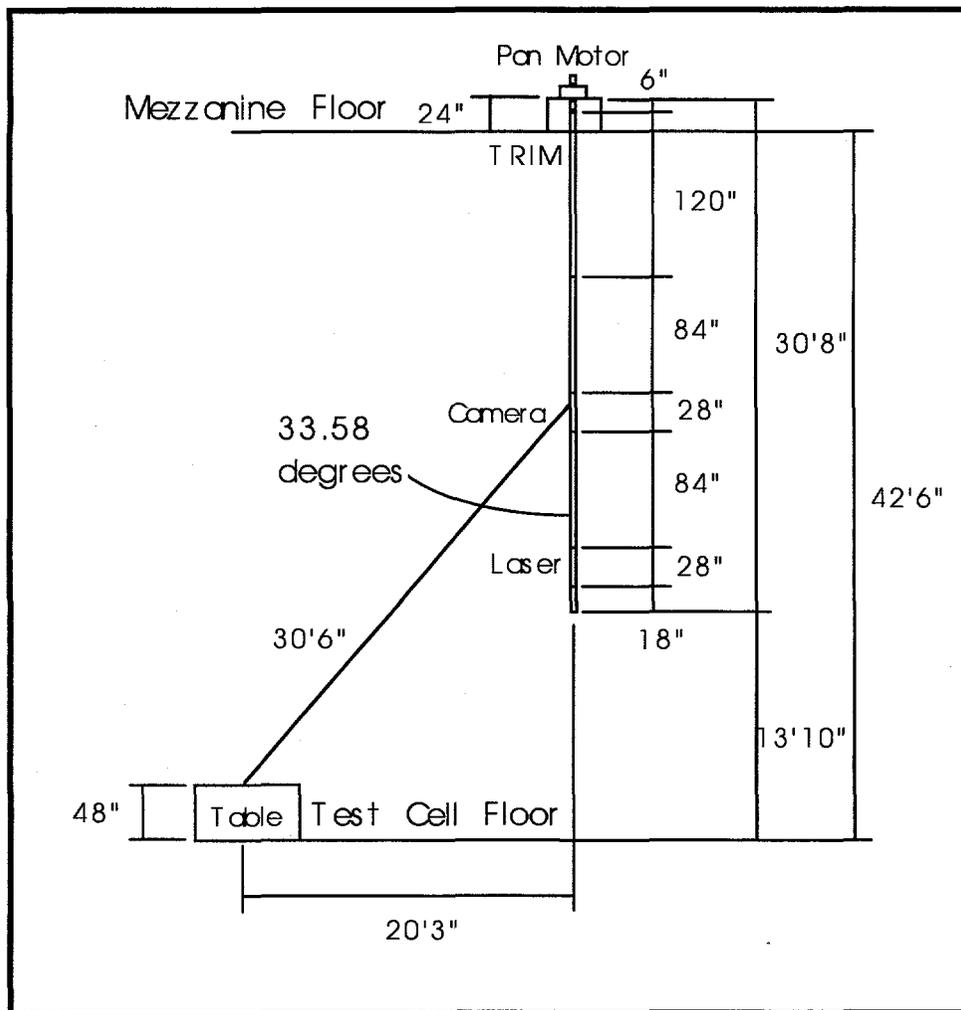


Fig. 5. TMS deployed in the FMEF cold test cell.

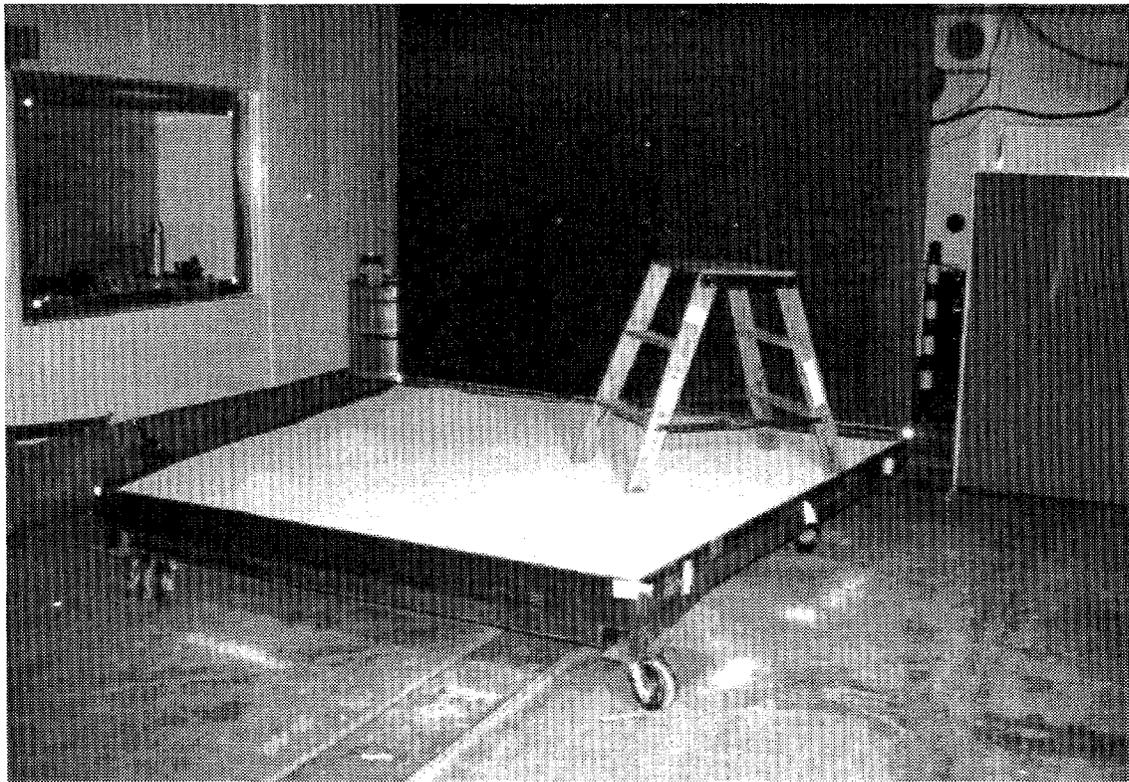


Fig. 6. Sandbox with mounds 1 through 4 used for volumetric analysis.

Table 1. Volumetric analysis results

Test/description	Height (in.)	Container measurement (in. ³)	TMS measurement (in. ³)	Accuracy (%)
Table top	2.2	554.7	514.6	93
Mound 1	4.42	1109.4	894.27	81
Mound 2	2.68	221.88	165.96	75
Mound 3	1.74	110.94	85.29	77
Mound 4	4.92	1553.16	1340.3	86

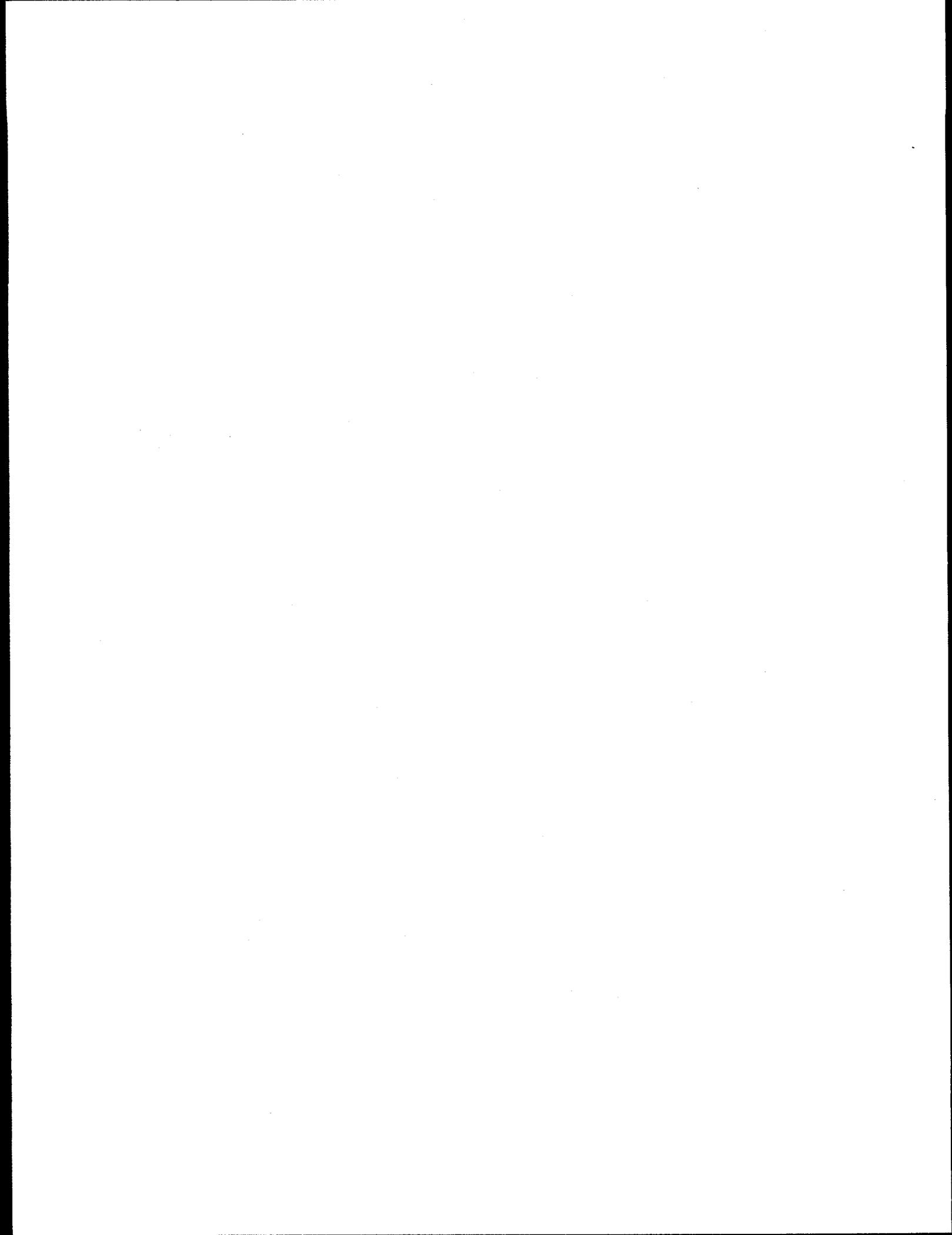
4. FUTURE WORK

More work is required on the TMS system before it can be used effectively to perform functions such as measuring the volume of waste in a UST or building a 3-D model for robotic path planning. Three areas need work. First, the laser-pointing system needs further calibration to improve the absolute accuracy of the system. When the calibration was performed at ORNL, it was believed that if the misalignment in the laser pointing could be carefully characterized and then added to the kinematic model, the system would be able to operate within the required specifications of ± 0.64 cm (0.25 in.) at 13.7 m (45 ft). However, this characterization assumption has not proved to be true. Because of the interaction among the 45-degree pointing mirror, the rotary table, and the laser, the laser must be pointing perfectly horizontal when the rotary table is at zero degrees. After the laser is aligned and the system is calibrated, a full characterization needs to be performed to determine the system accuracy over its targeted measurement space. Second, the volumetric analysis capability needs to be added to the ICERVS tool. The present ORNL-developed tool can determine only the volumes for very structured environments with no vertical surfaces present. ICERVS's octree database enables it to easily deal with the unstructured environments of the USTs. In addition, the identification and elimination of objects in the UST that should not be in the volumetric calculation of waste can easily be achieved by functions within the ICERVS tool. Third, the error model that indicates the confidence of the measurement based on (1) range, (2) contrast, (3) signal-to-noise ratio, (4) camera sensitivity, (5) laser power, (6) angle of incidence, (7) kinematics of the sensor head, (8) temperature, and (9) radiation needs to be developed, calibrated, and empirically verified. Currently, the TMS has been characterized along one horizontal plane extending out the Y axis of the sensor in the home position. The characterization needs to be extended to include multiple such characterizations on horizontal planes every 1 m (3.2808 ft) over the entire measurement space. This would allow arrays of characterization data to be formulated for axial range, lateral range, and angle of incidence over the targeted measurement space. In addition, more data need to be gathered on the TMS's capability to map surfaces of varying texture and therefore reflectance and absorption properties. For the present characterization, most of the measurements were performed only once. A subset of the data points needs to be collected multiple times in an effort to gather statistics on the system's repeatability.

In addition, there are a few improvements that could be made to the system that would greatly improve its versatility as well as its deployability. A set of lights distributed about the sensor head would greatly aid in tank inspection as well as installation and removal through the riser. Lights could also be added to the extension modules or the dummy ESS. This would allow the lights to be designed for decontamination as opposed to adding lights to the outside of the sensor, which further restricts the aperture of the penetration. In addition, a zoom camera with a dedicated pan and tilt mechanism could be added to the bottom of the ESS.

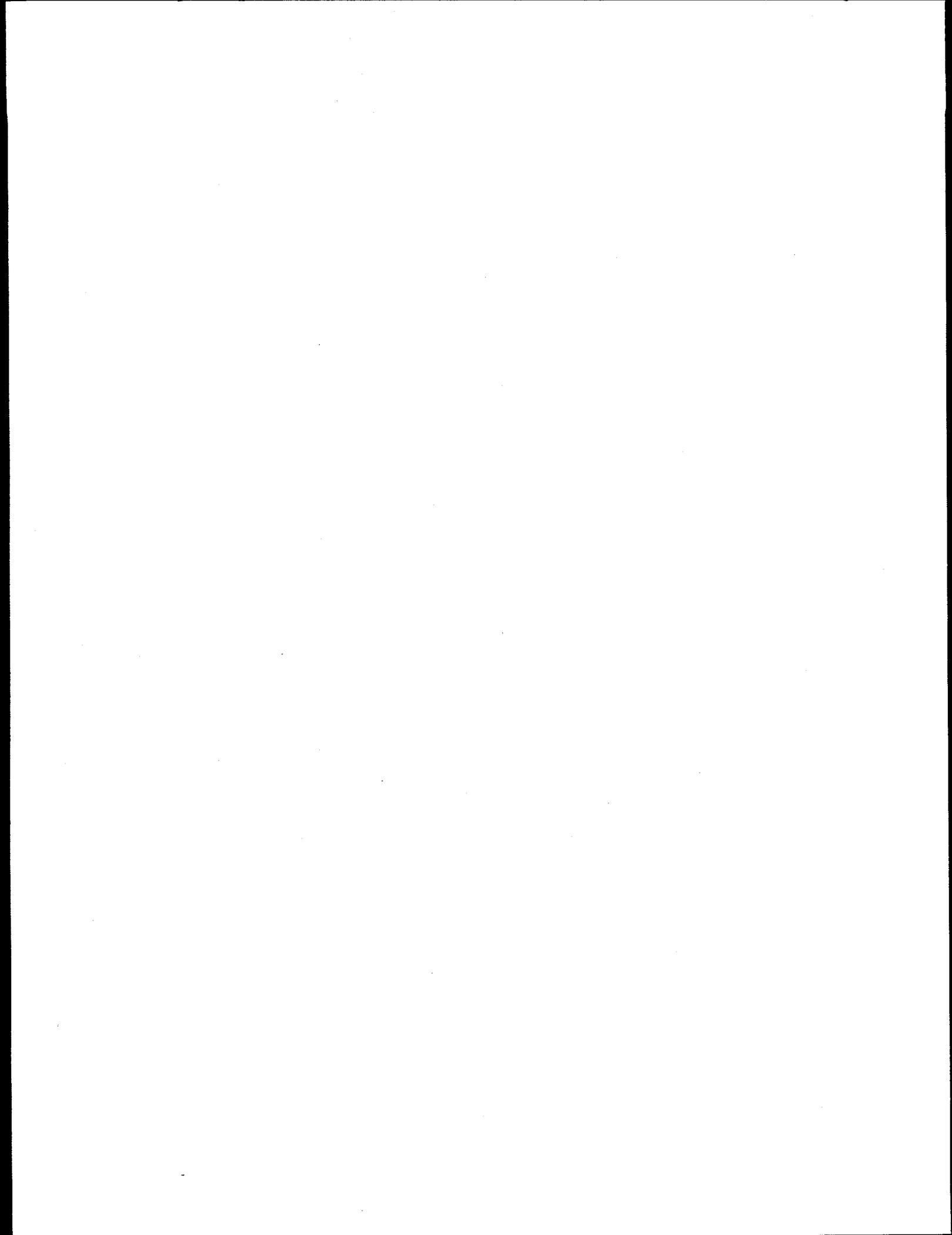
A strong-back and trailer could be used to deploy the TMS sensor head. The strong-back would be approximately 10.67 m (35 ft) in length. The strong-back would support the TMS in a lateral position and would allow the TMS to be configured on its side and then lifted with a crane to a vertical position. This would eliminate the necessity to top-down assemble the TMS sensor head with a crane. The strong-back could be enclosed in a 10.67-m (35-ft) trailer that would also serve as a containment box for transporting the TMS sensor head among deployment sites.

A control trailer could be purchased to house the control station for the TMS. An additional trailer could serve as a maintenance trailer for repairing and performing limited recalibration of the TMS camera and laser modules in the field.



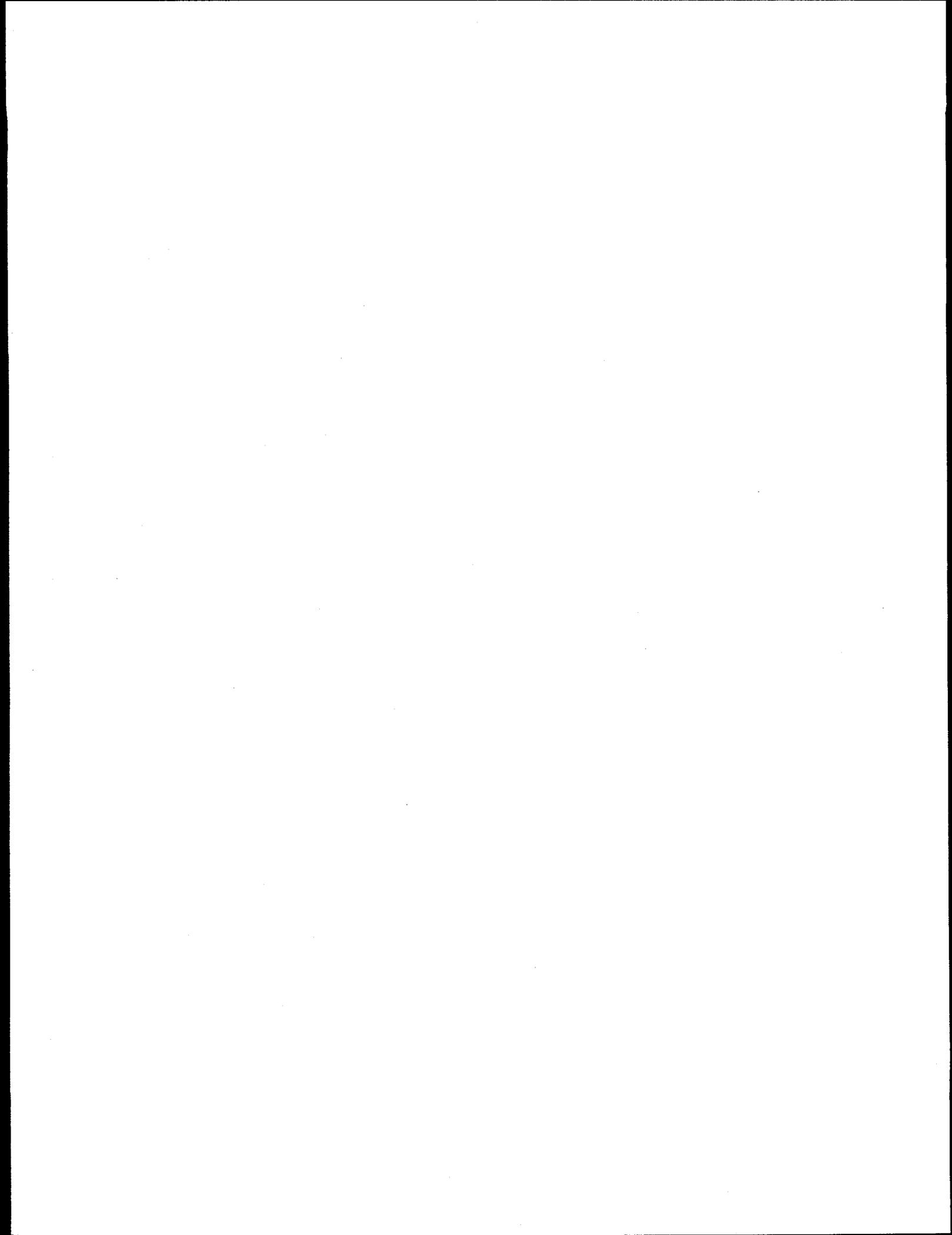
5. FUTURE PLANS

A proposal has been written by PNNL and ORNL for deploying the TMS in UST AX104 at Hanford during FY 1998 to measure the remaining waste left in the tank. The TMS is also being considered for deployment in UST CY106 to measure the effectiveness of vendor-supplied sluicing equipment.



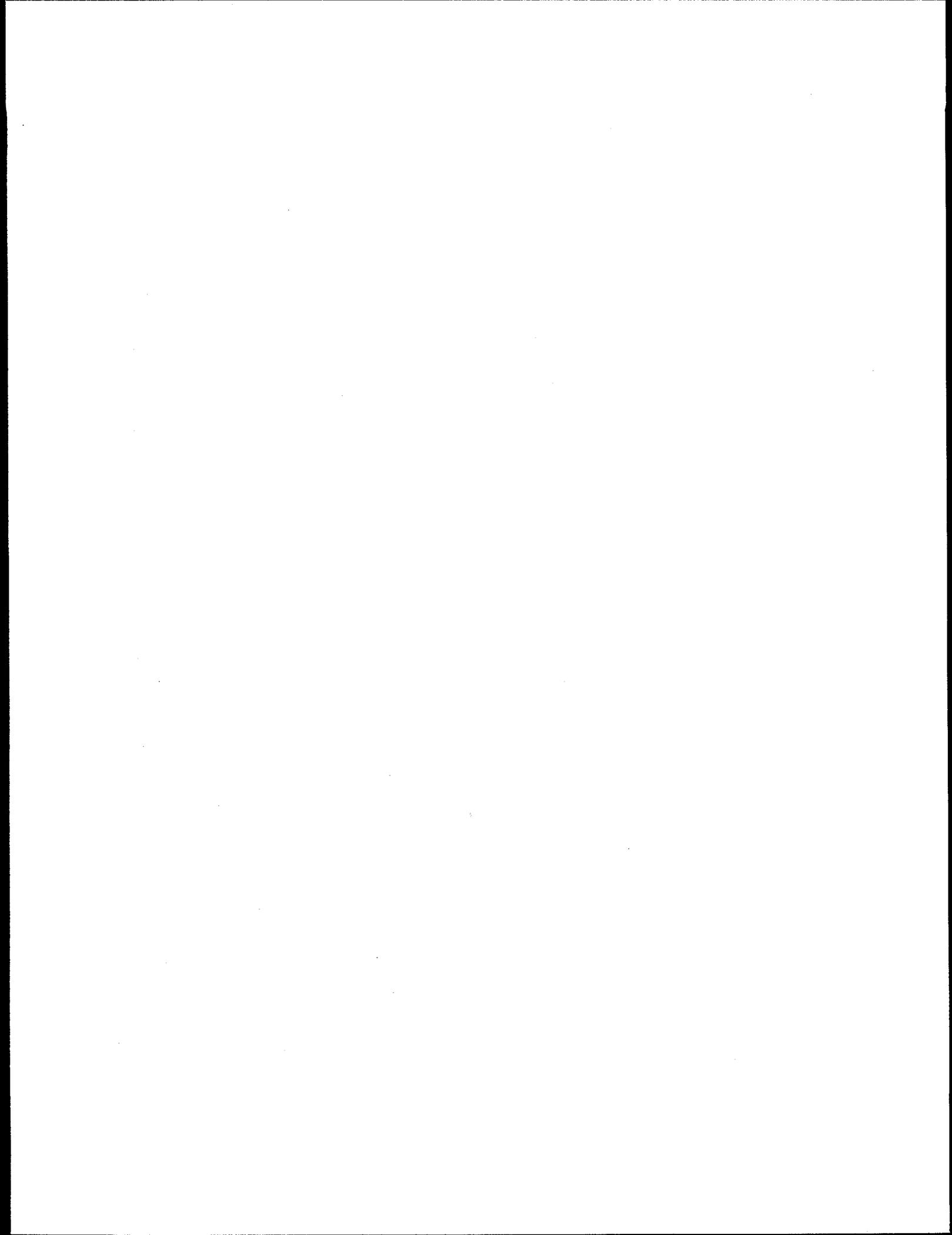
6. REFERENCES

1. W. W. Jenkins, *Remote Handling Equipment for the Removal of Waste from Single-Shell Tanks at the Hanford Site*, WHC-SA-0934-FP, Westinghouse Hanford Company, Pacific Northwest Natl. Lab, 1990.
2. B. L. Burks, et al., *Waste-Surface Mapping of the Fernald K-65 Silos Using a Structured Light Measurement System*, ORNL/TM-12185, Martin Marietta Energy Systems, Inc., Oak Ridge Natl. Lab., 1992.
3. G. A. Armstrong, B. L. Burks, and D. V. Hoesen, *South Tank Farm UST Inspection using the Topographical Mapping System for Radiological and Hazardous Environments*, ORNL/TM-13437, Lockheed Martin Energy Research Corp., Oak Ridge Natl. Lab., 1997.
4. J. C. Fulton, *Single Shell Tank Waste Retrieval Study*, WHC-EP-0352, Westinghouse Hanford Company, Pacific Northwest Natl. Lab, 1992.
5. S. A. Krieg, et al., *Riser Configuration Document for Single Shell Tanks*, WHC-SD-RE-TI-053, Westinghouse Hanford Company, Pacific Northwest Natl. Lab, 1990.
6. B. L. Burks, B. E. Bernacki, and G. A. Armstrong, "Topographical Mapping System for Radiological Environments," Statement of Work, Martin Marietta Energy Systems, Inc., Oak Ridge Natl. Lab., 1993.
7. C. M. Smith, *Functions and Requirements for the Light-Duty Utility Arm Integrated System*, WHC-SD-TD-FRD-003, Rev. 0, Westinghouse Hanford Company, Pacific Northwest Natl. Lab, March 1994.
8. Mechanical Technology, Inc., *Interactive Computer-Enhanced Remote-Viewing System (ICERVS)*, Final Report, November 1994–September 1996.
9. Pentax PTS-V Series, *Electronic Total Station Instruction Manual*, ASAHI Precision Co., Ltd., 1994.
10. Guo-Qing Wei and Song De Ma, *Two Plane Camera Calibration: A Unified Method*, National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 1991.
11. F. W. DePiero, "Camera Calibration in a Hazardous Environment Performed In Situ with Automated Analysis and Verification," Martin Marietta Energy Systems, Inc., Oak Ridge Natl. Lab., Fifth Annual ANS Topical Meeting, 1993.

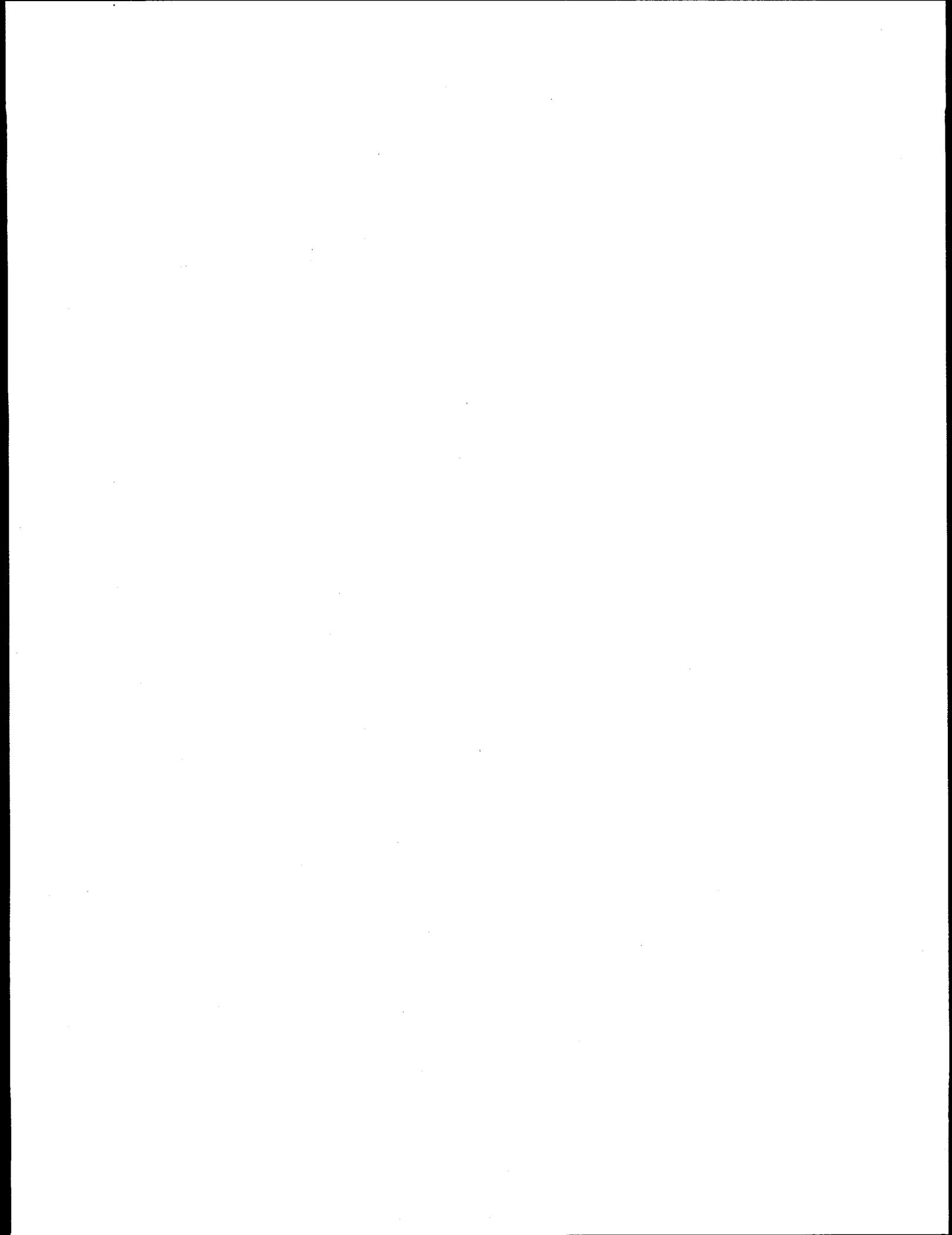


7. SELECTED BIBLIOGRAPHY

1. G. A. Armstrong et al., "Topographical Mapping System for Hazardous and Radiological Environments," ANS Winter Meeting, October 1995.
2. R. Gamache, "A Topological Mapping System for Radiological Environments," Mobile Mapping Symposium, Ohio State University, Columbus, Ohio, May 24-25, 1995.
3. J. Wagner, "Mapping of Waste Surfaces," DOE Conference on Environmental Commerce, Chattanooga, Tenn., October 1993.
4. J. Tourtellott, "Visualization and Modeling of 3-D Image Data in Remote Robotic Applications," *SPIE Proceedings on Intelligent Robots and Computer Vision*, Philadelphia, Vol. 2588, October 25-28, 1995.
5. Tourtellott and J. Wagner, "Interactive Computer Enhanced Remote Viewing System," DOE Conference on Partnering, Morgantown W.Va., October 1995.



APPENDIX A
TMS SURFACE MAPS



Volumetric Analysis of Surface map data from the TMS

Table Top w/ Sand Map

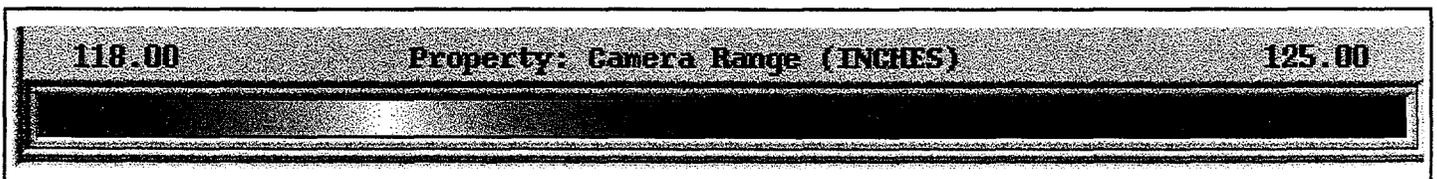
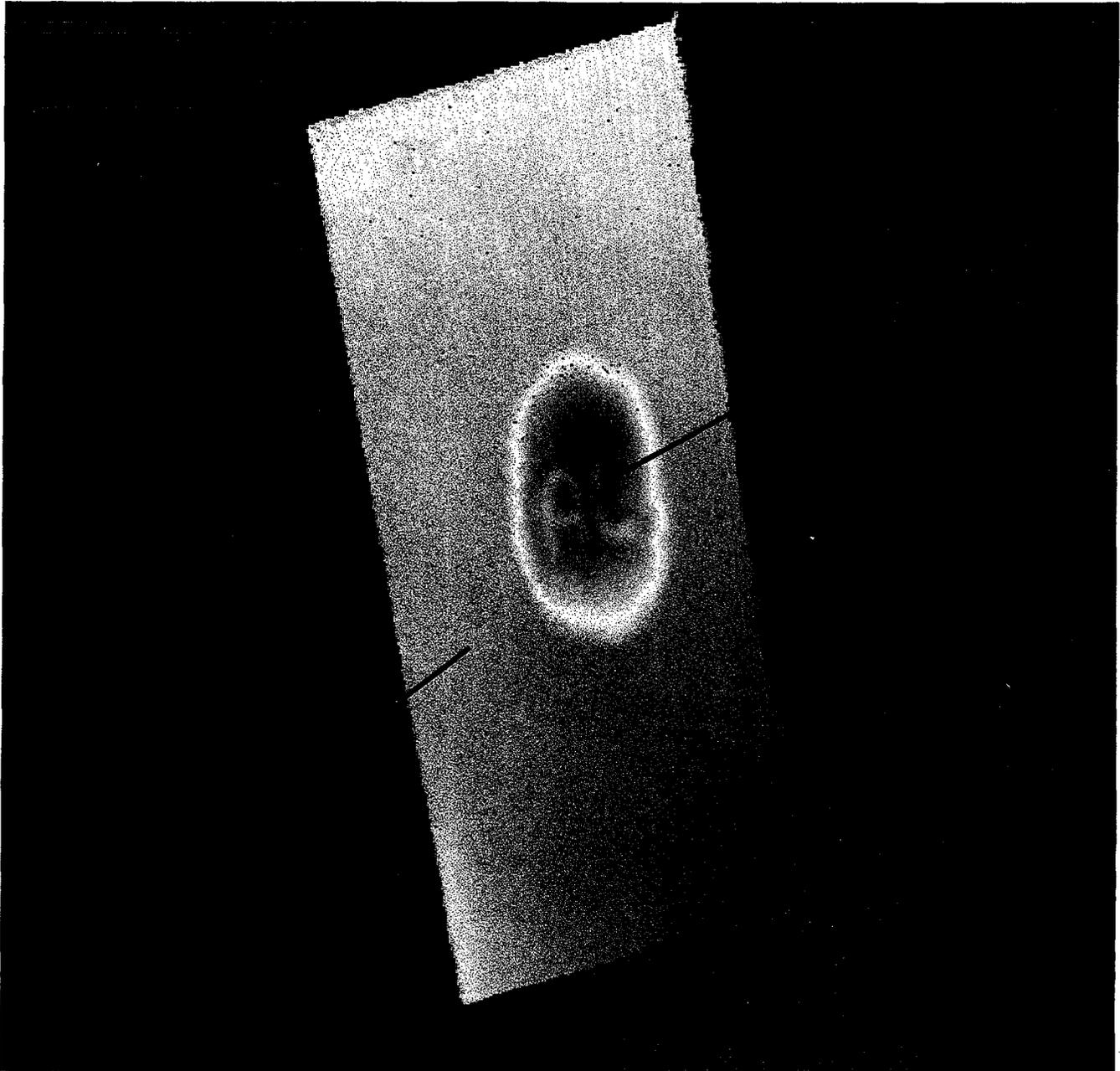
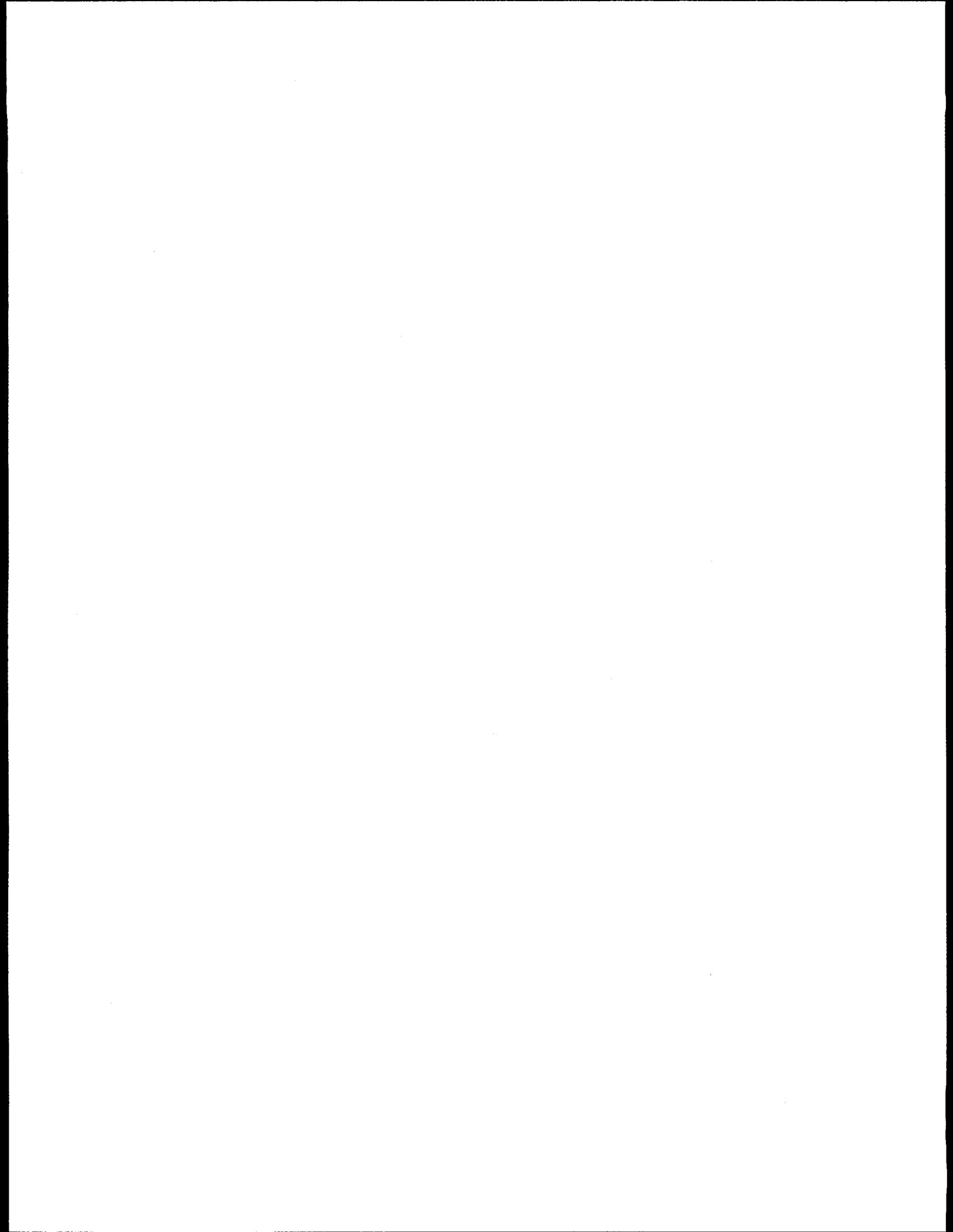


Figure A.1. Table top surface map with sand mound.



Volumetric Analysis of Surface map data from the TMS

Table Top Difference Map



VOLUMETRIC RESULTS

Actual = 554.7 cubic inches

Measured = 514.6 cubic inches

Accuracy 93%

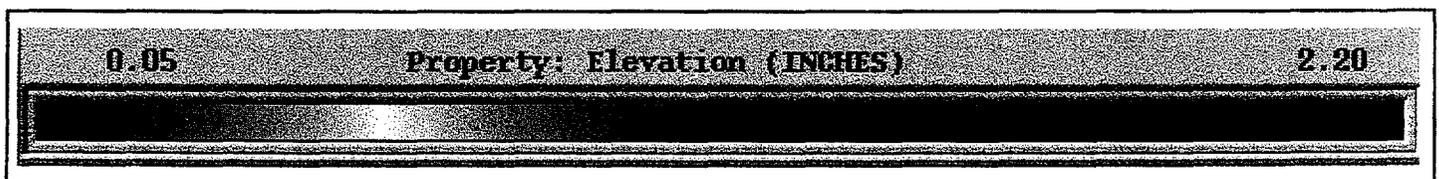
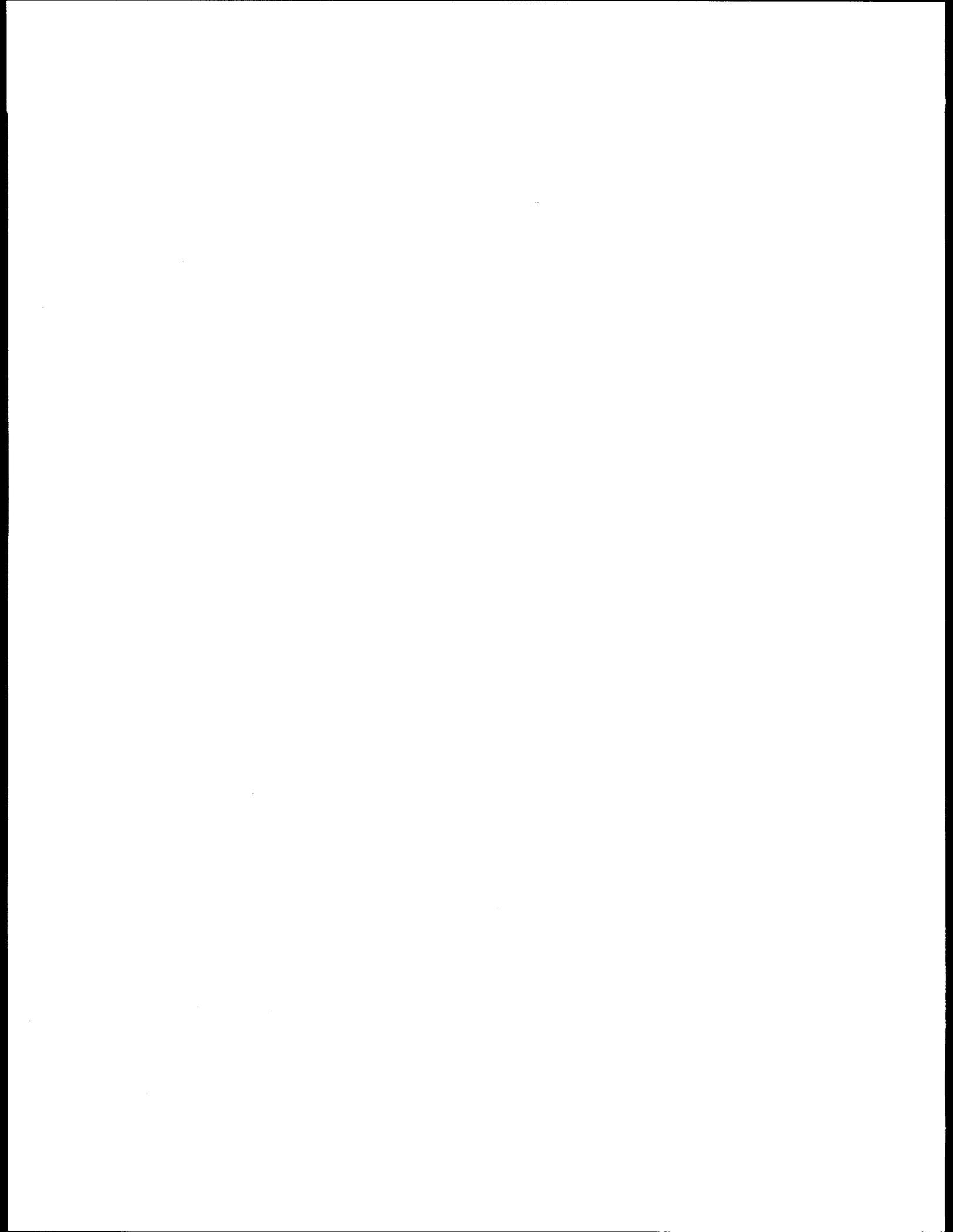


Figure A.2. Table top difference map.



Volumetric Analysis of Surface map data from the TMS

Left Sandbox – Mounds 1–3

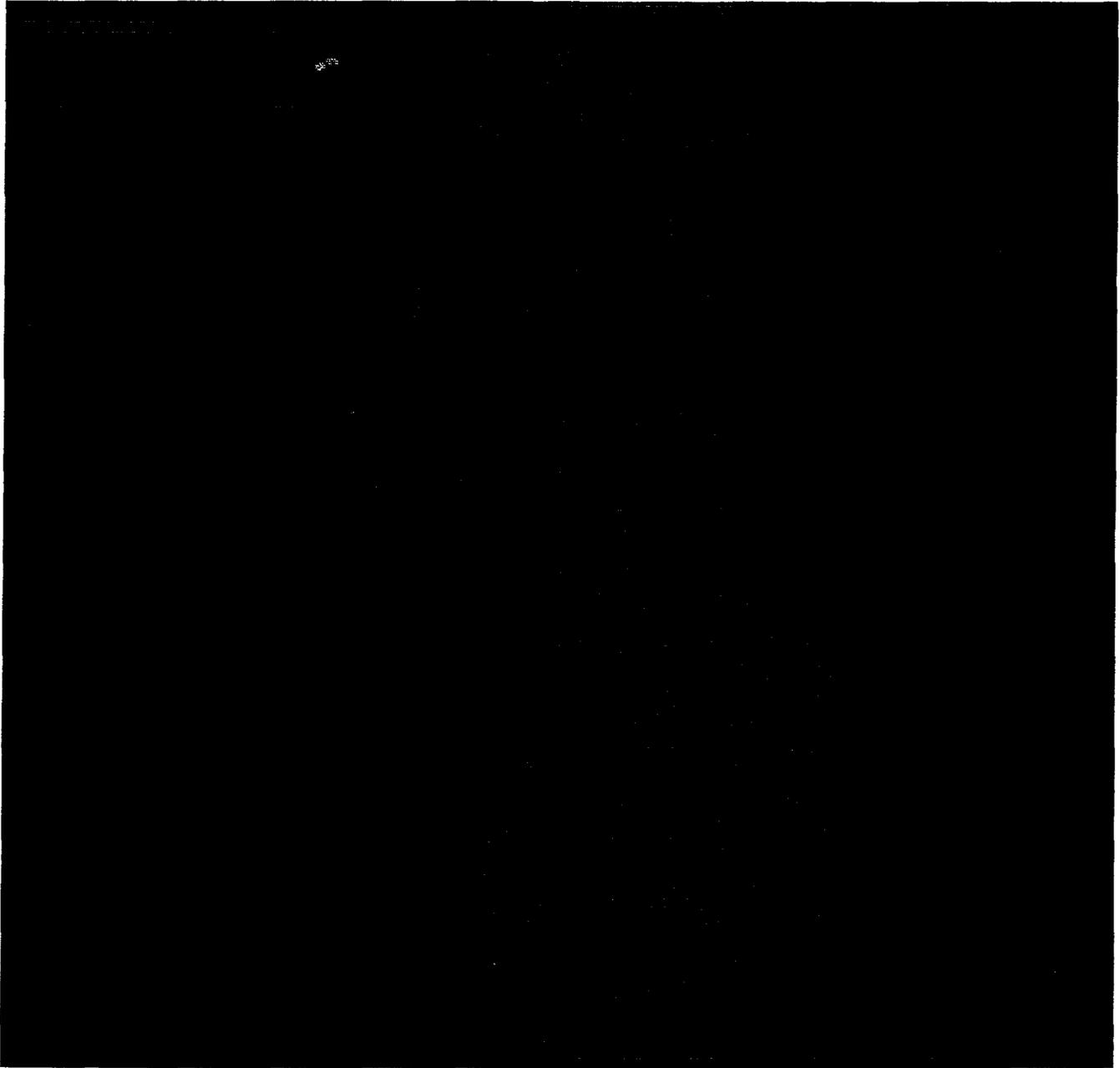
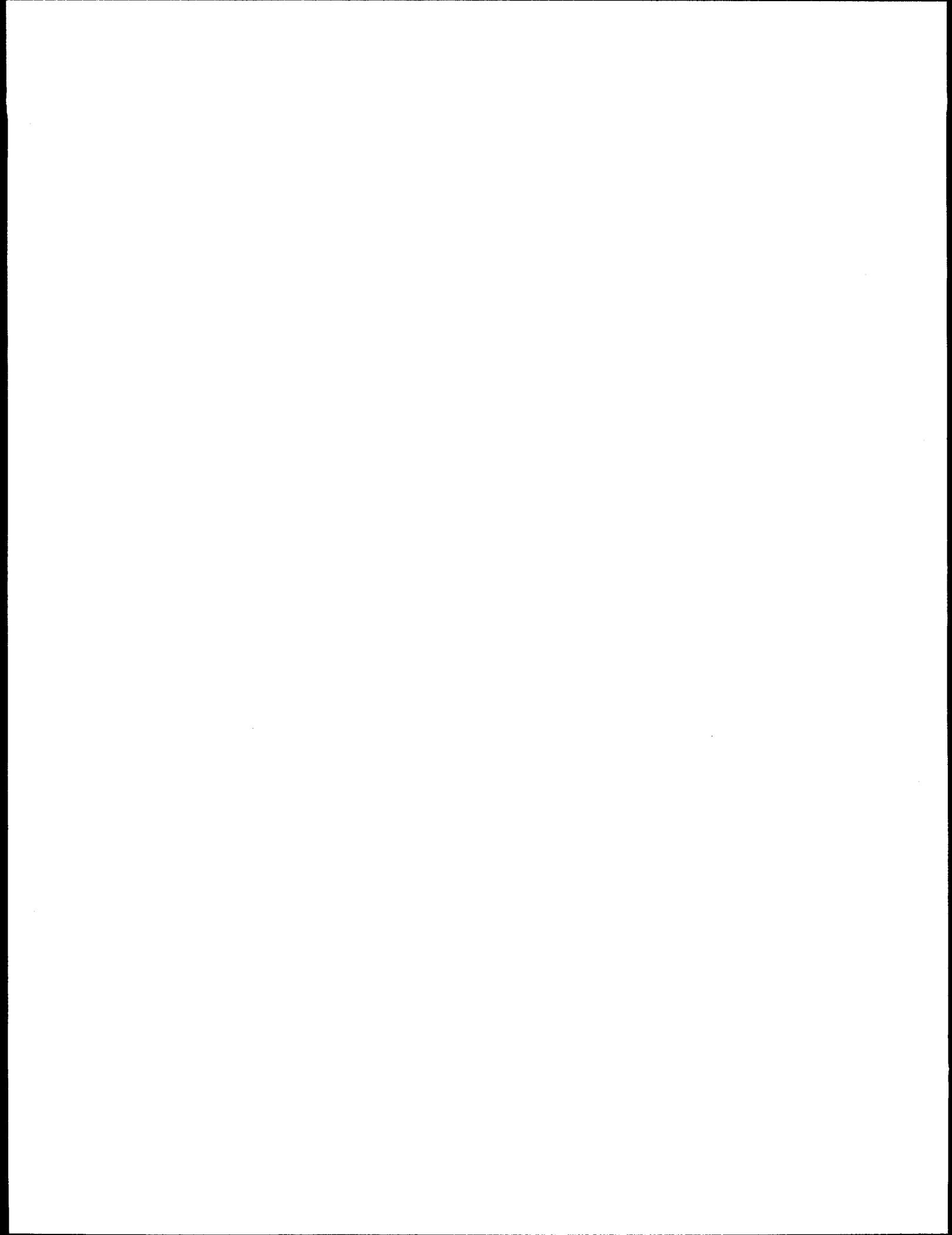
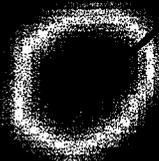
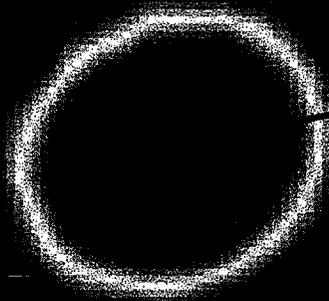


Figure A.3. Left sandbox surface map with mounds 1, 2, and 3.



Volumetric Analysis of Surface map data from the TMS

Sandbox Left Difference Map



VOLUMETRIC RESULTS

Mound	Actual [cubic inches]	TMS [cubic inches]	%
1	1109.4	894.27	81
2	221.88	165.96	75
3	110.94	85.29	77

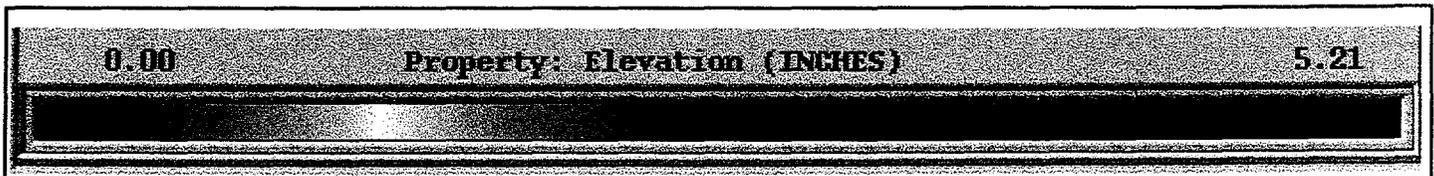
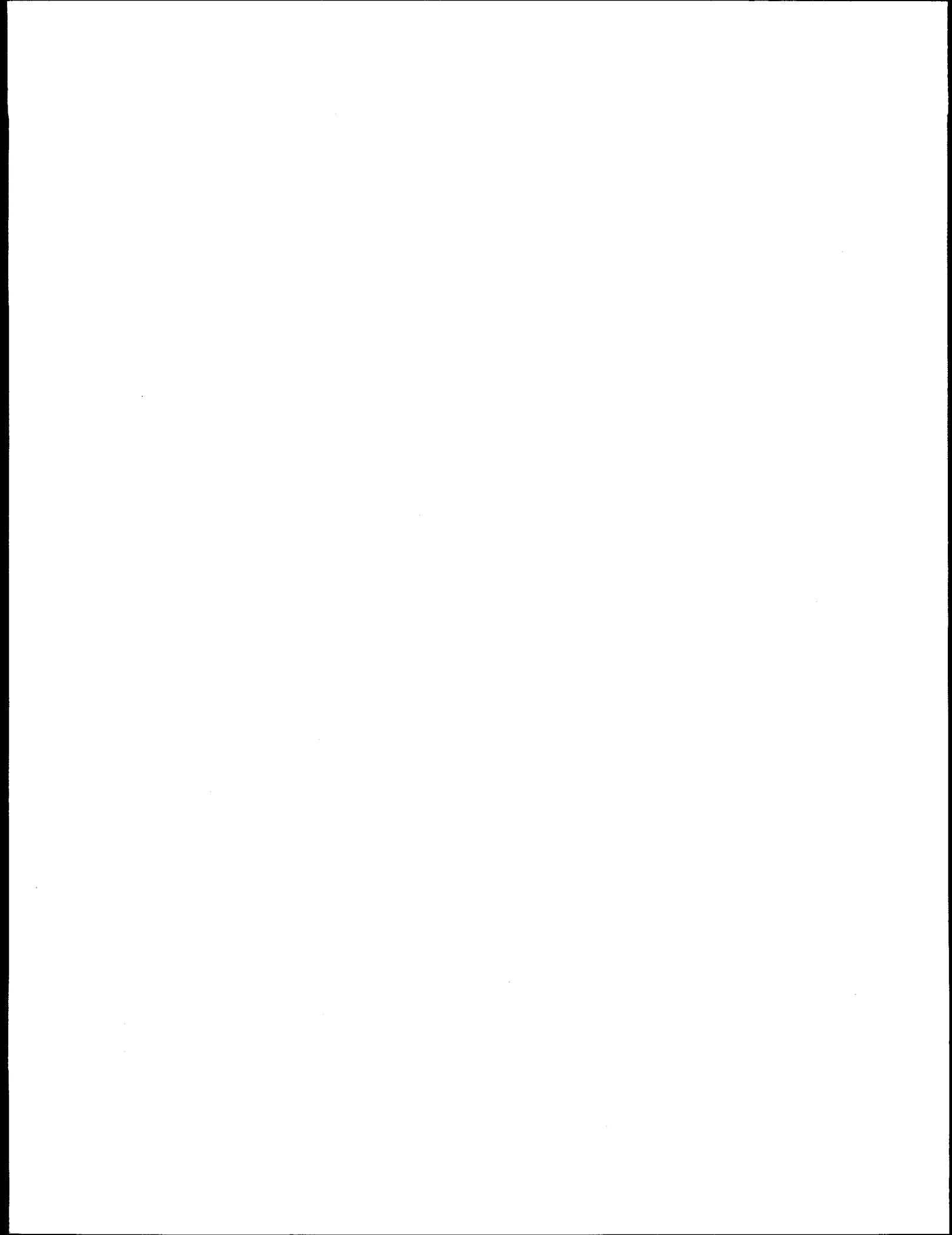


Figure A.4. Left sandbox difference maps of mounds 1, 2, and 3.



Volumetric Analysis of Surface map data from the TMS

Right Sandbox – Mounds 4

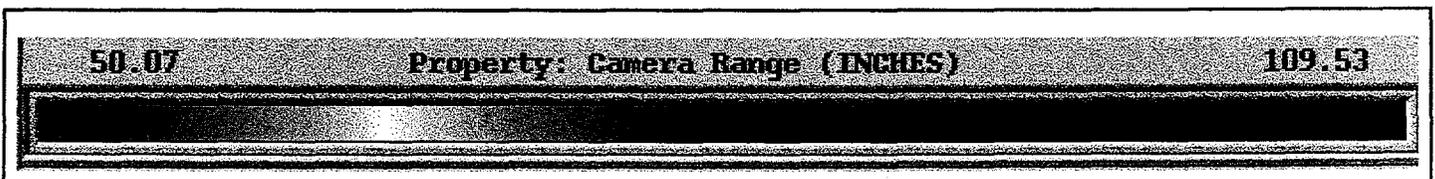
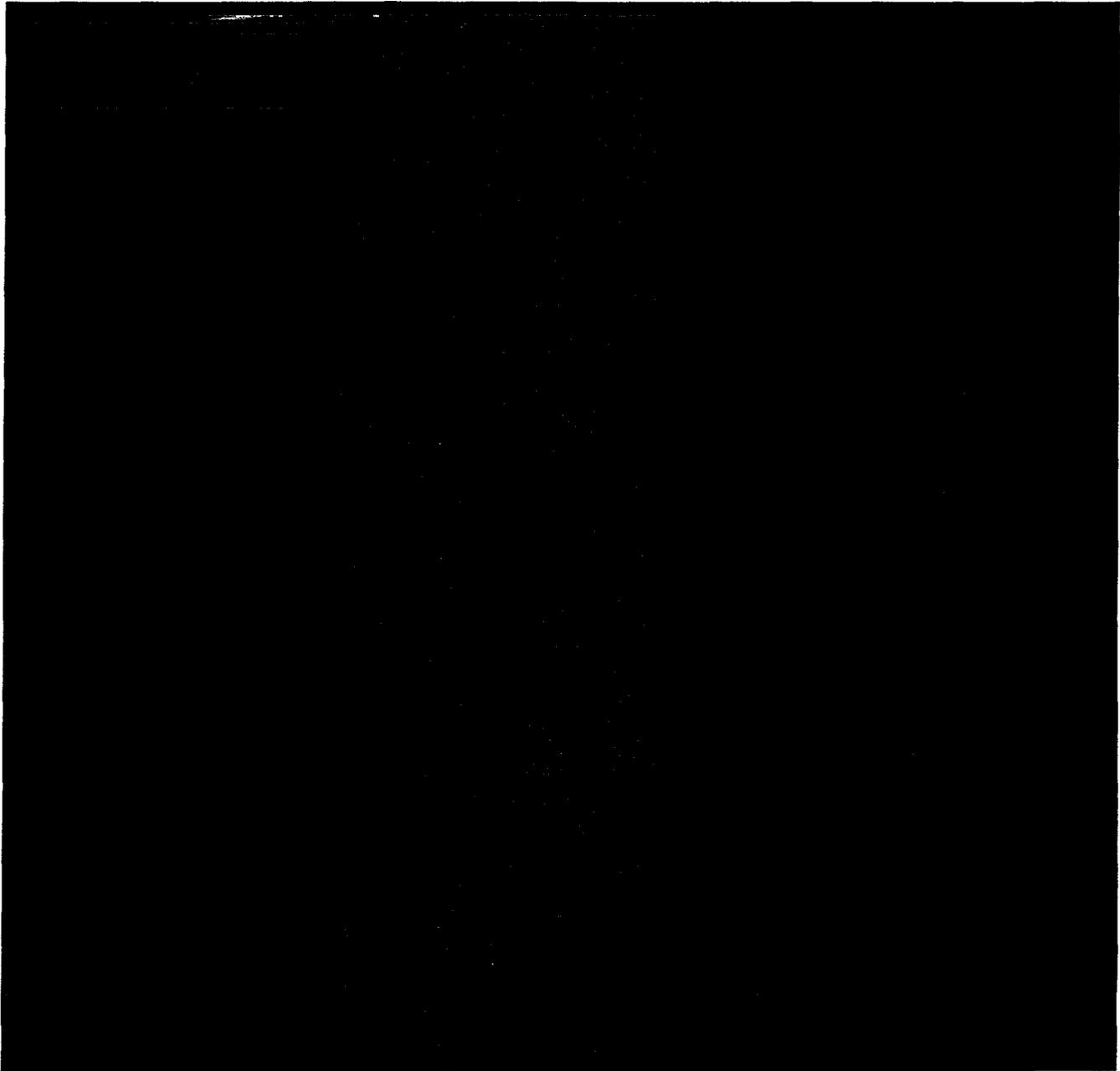
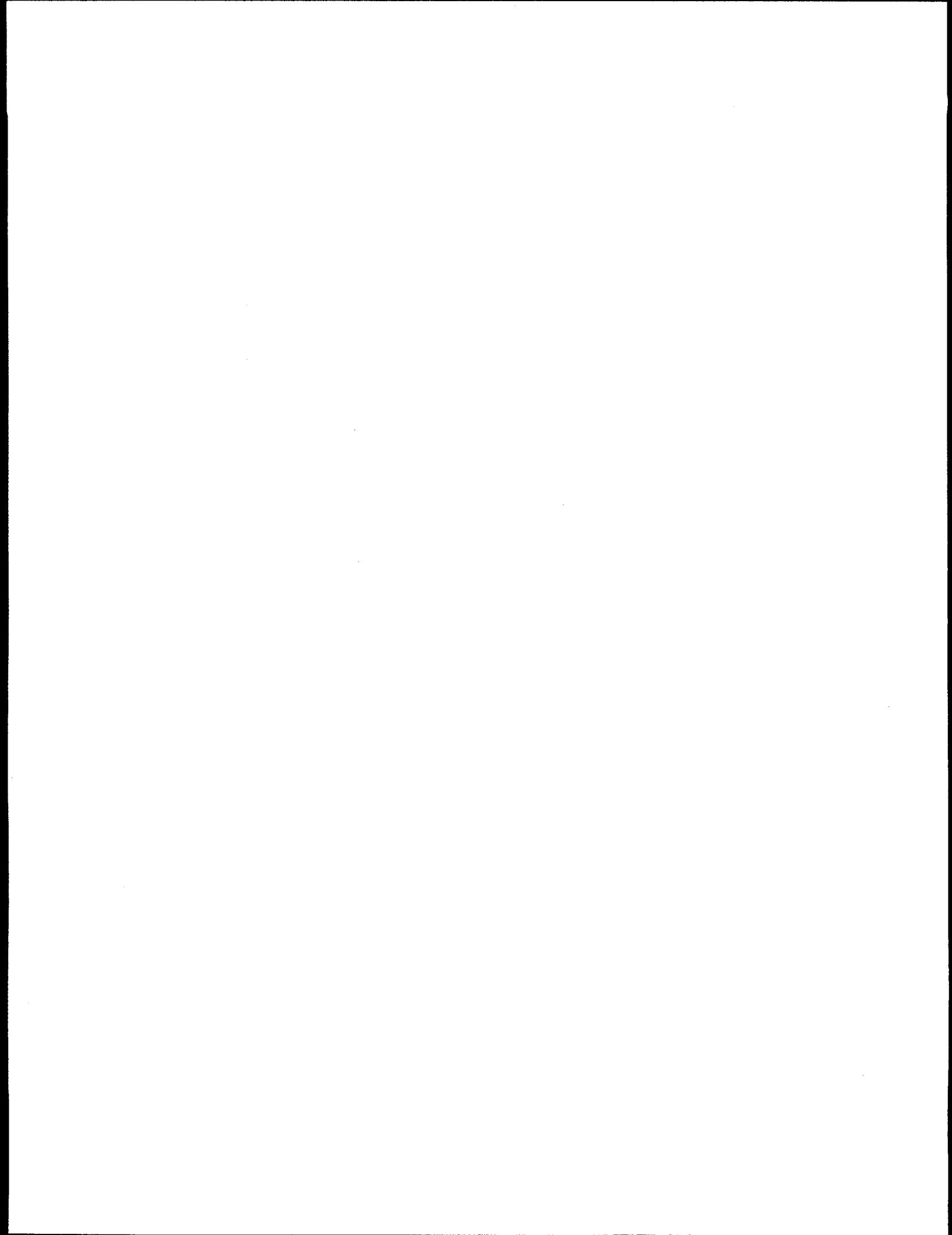
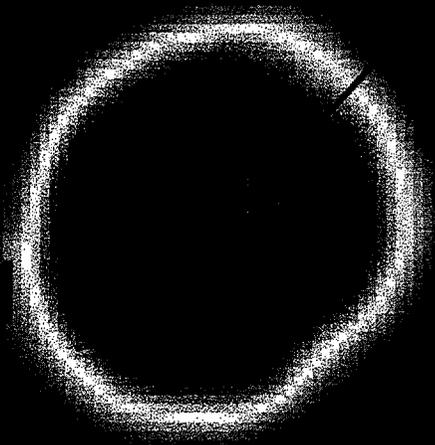


Figure A.5. Right sandbox surface map with mound 4.



Volumetric Analysis of Surface map data from the TMS

Sandbox Right Difference Map



VOLUMETRIC RESULTS

Actual = 1553.16 cubic inches

Measured = 1340.3 cubic inches

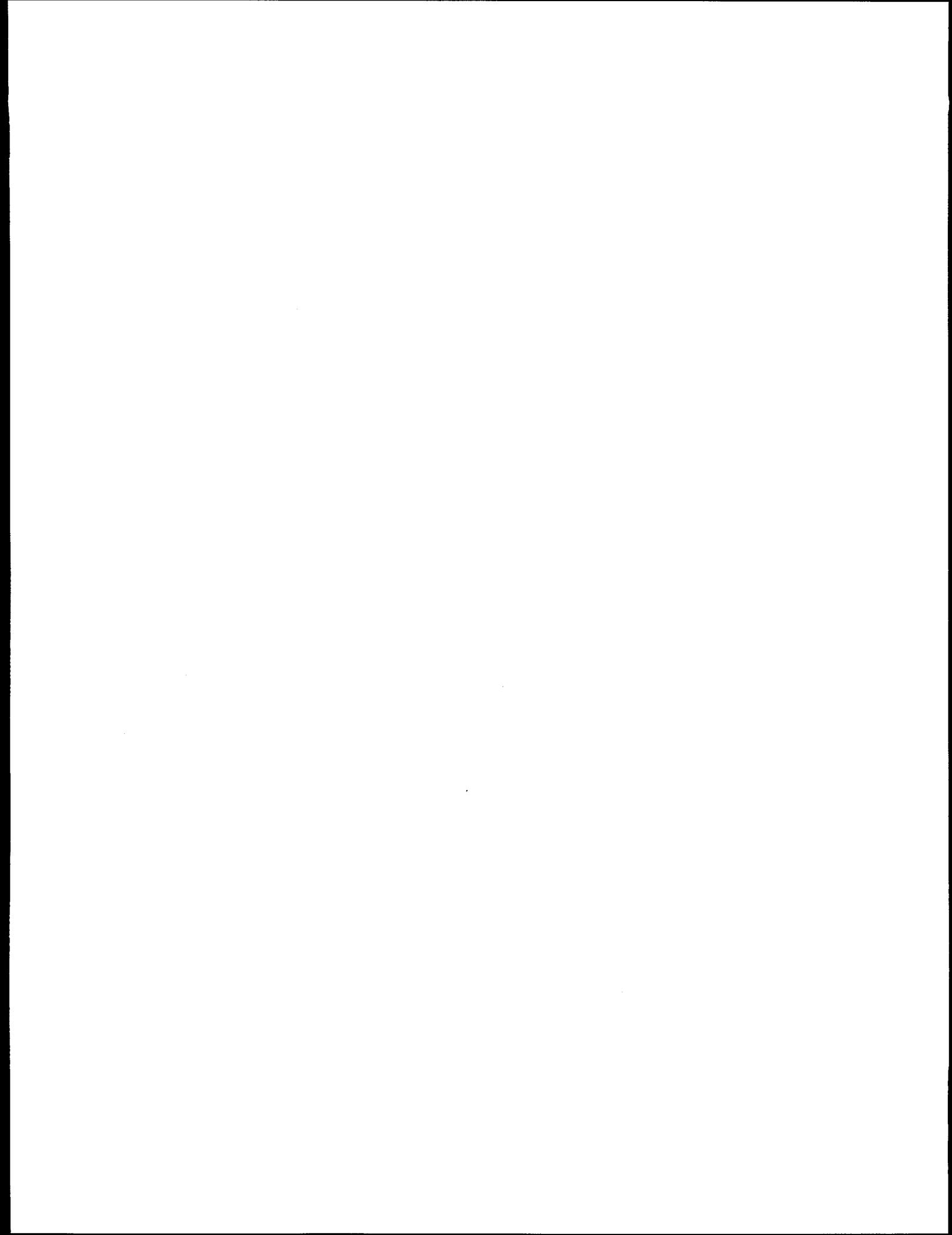
Accuracy 86 %

0.00

Property: Elevation (INCHES)

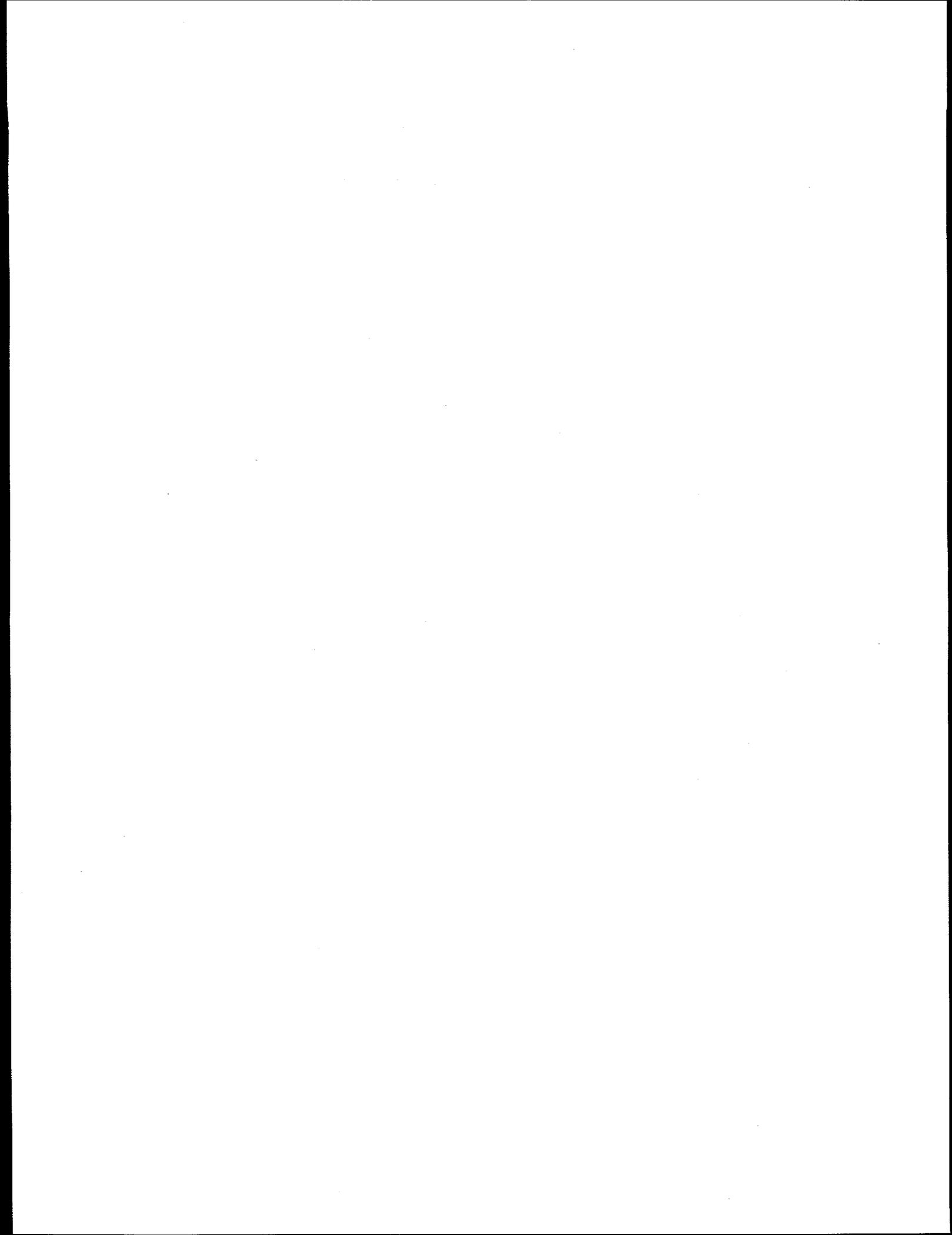
6.30

Figure A.6. Right sandbox difference map of mound 4.



APPENDIX B

TMS DIAGRAMS AND PHOTOGRAPHS



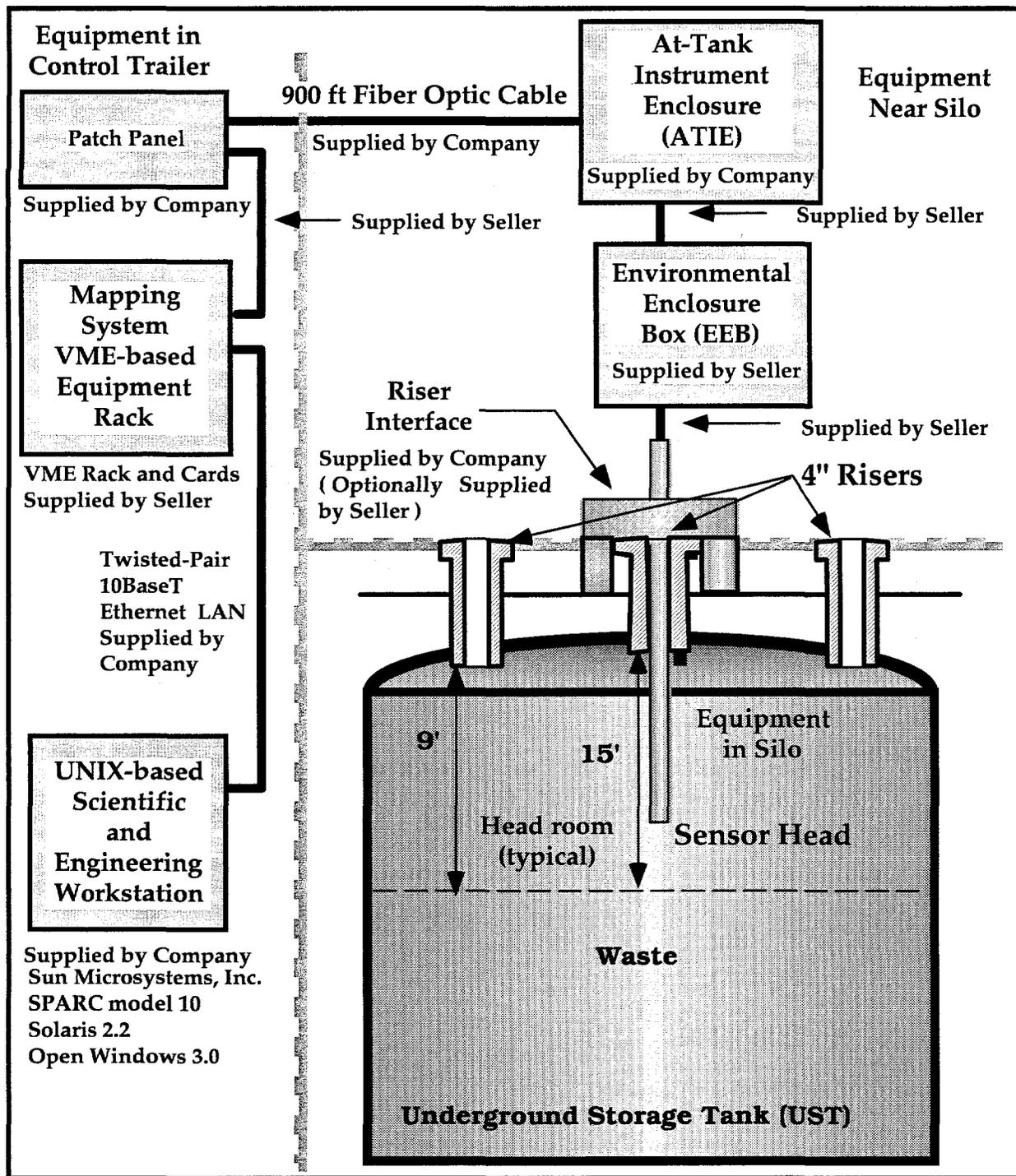


Fig. B.1. Block diagram of the TMS.

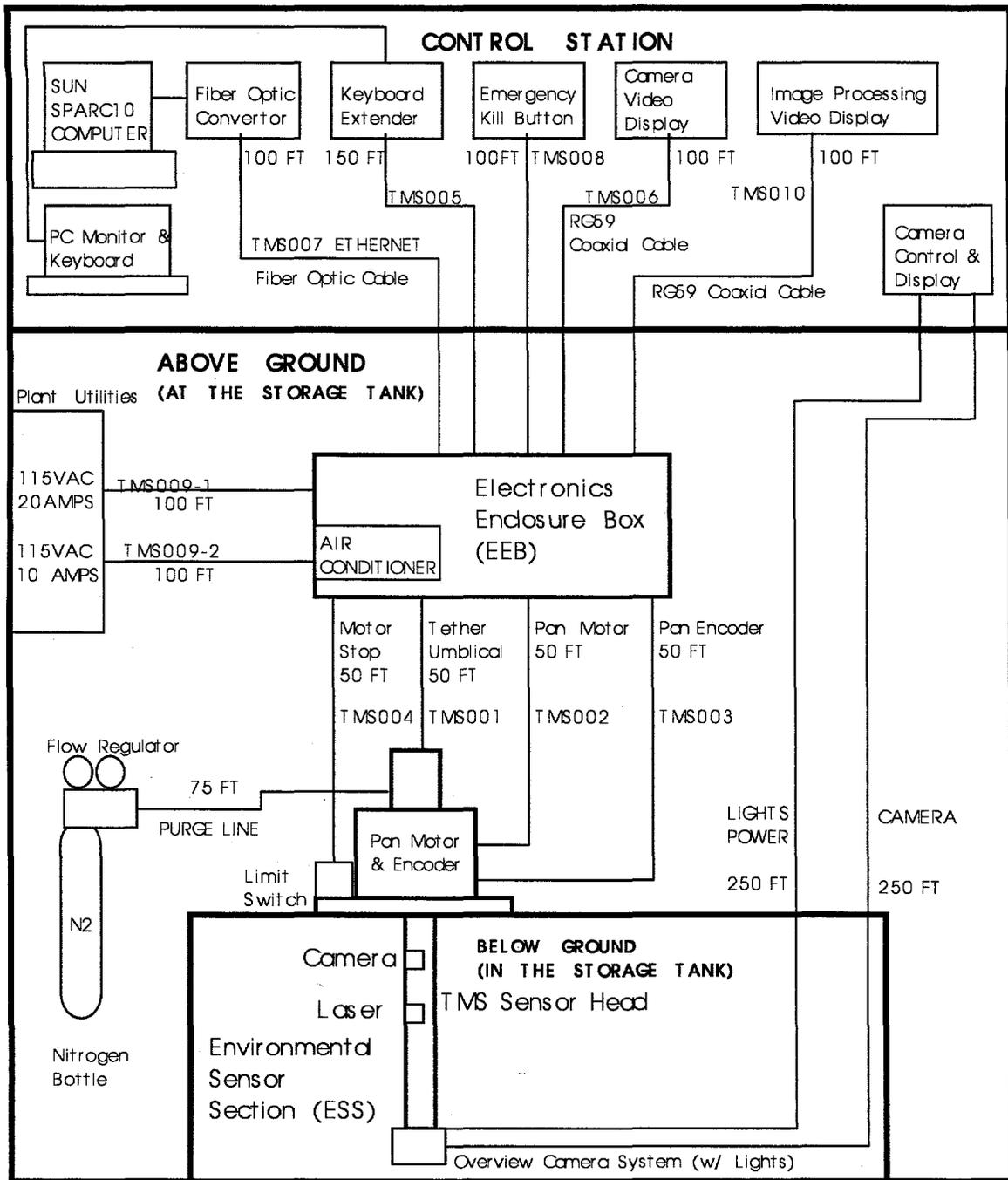


Fig. B.2. TMS cable diagram.

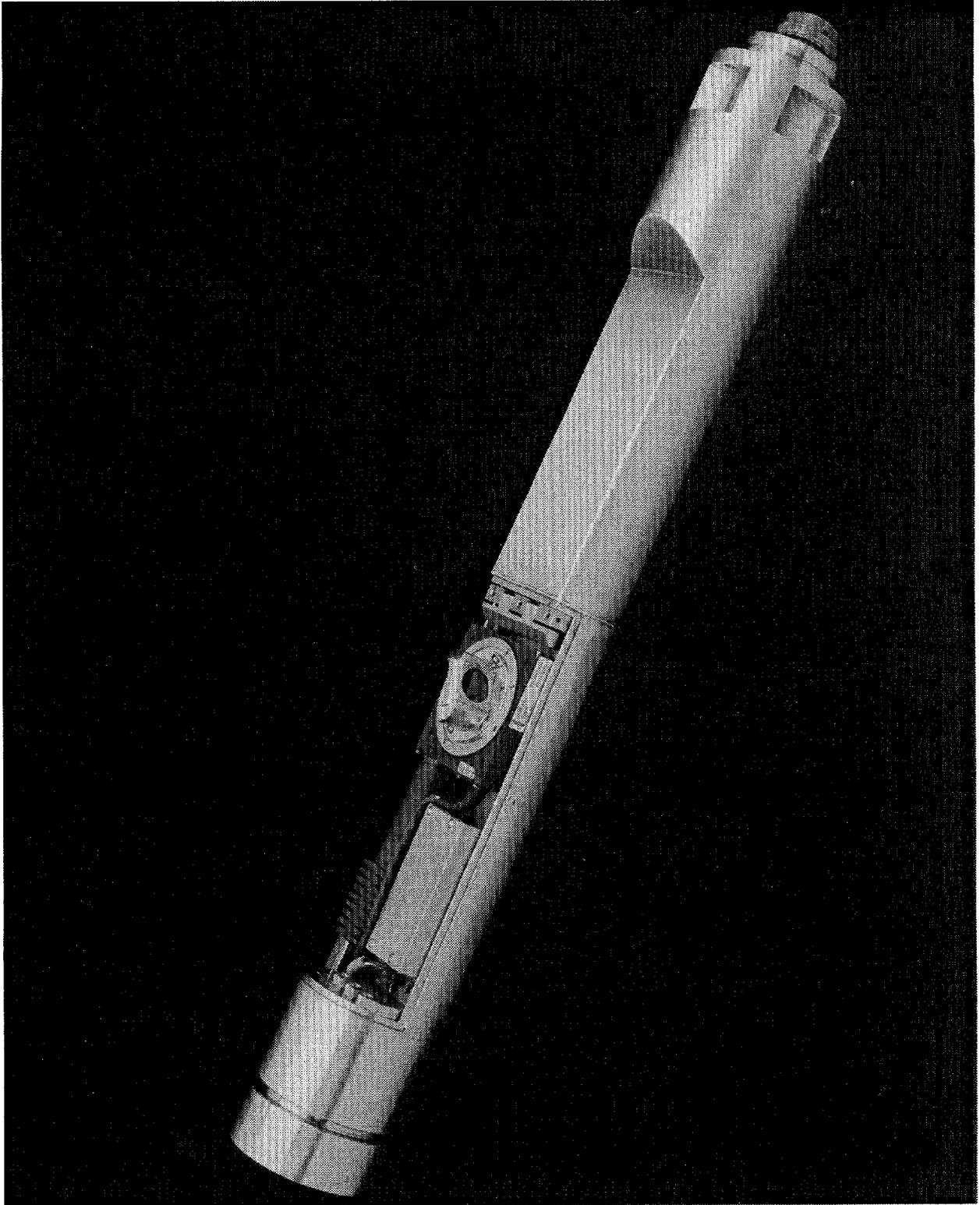


Fig. B.3. TMS laser/camera module.

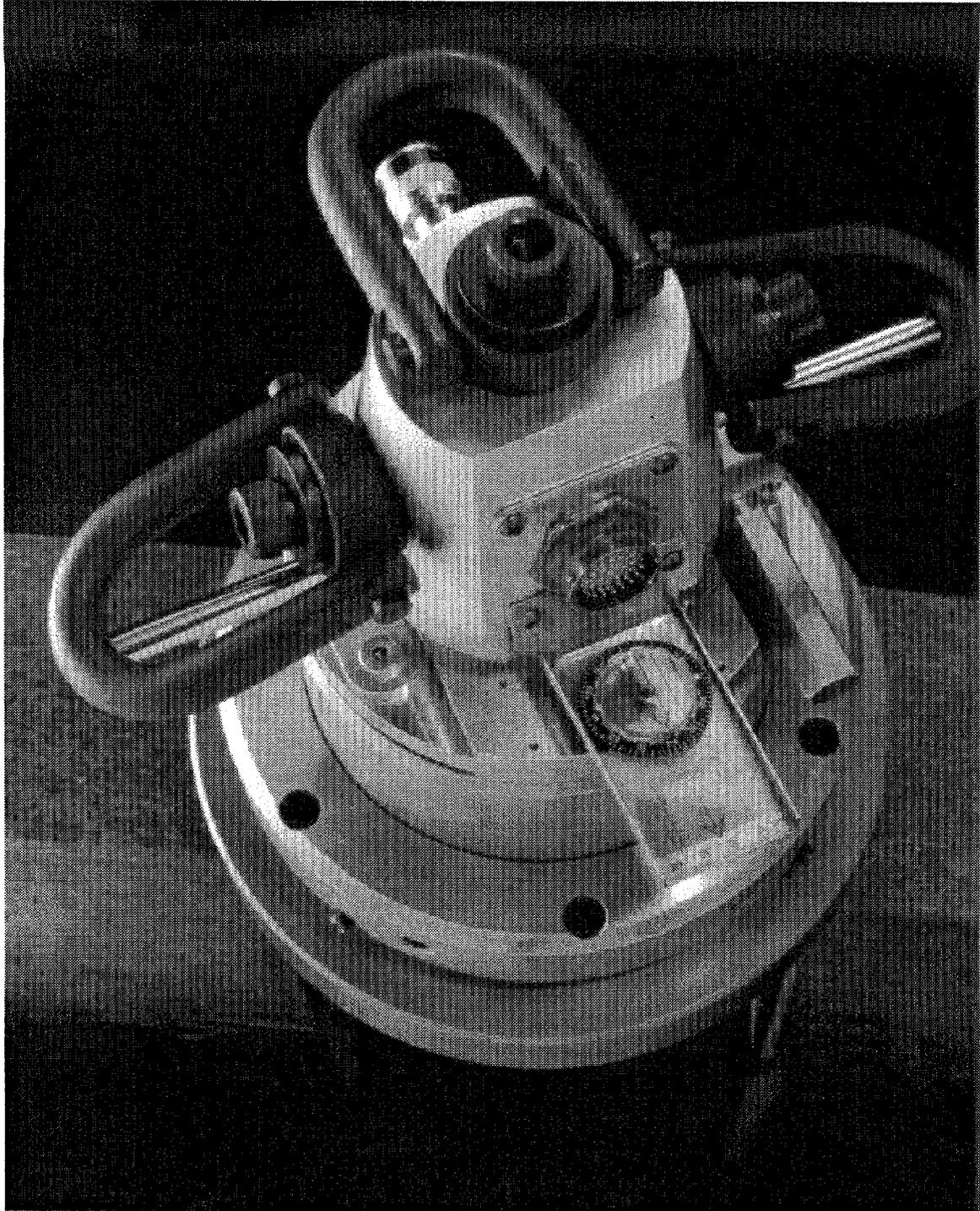


Fig. B.4. TMS pan motor adapter module.

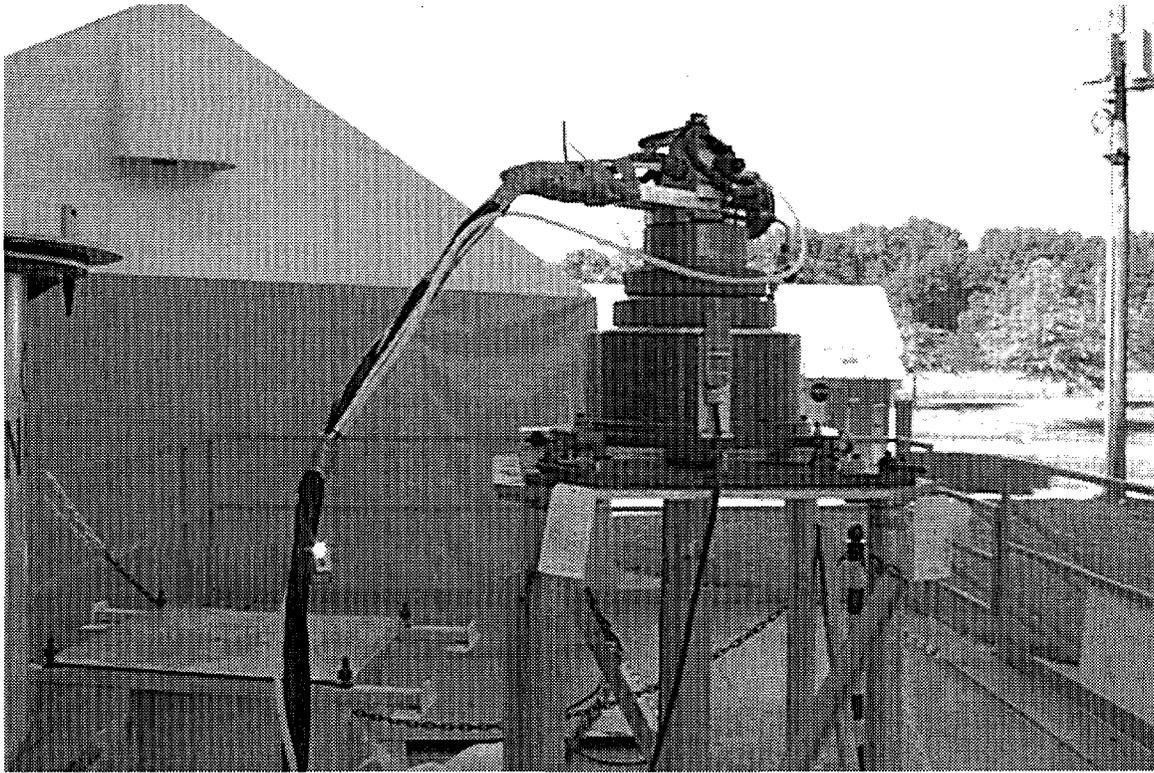


Fig. B.5. TMS deployed on the bridge of the Robotics and Process Systems Division's Technology Test Facility.

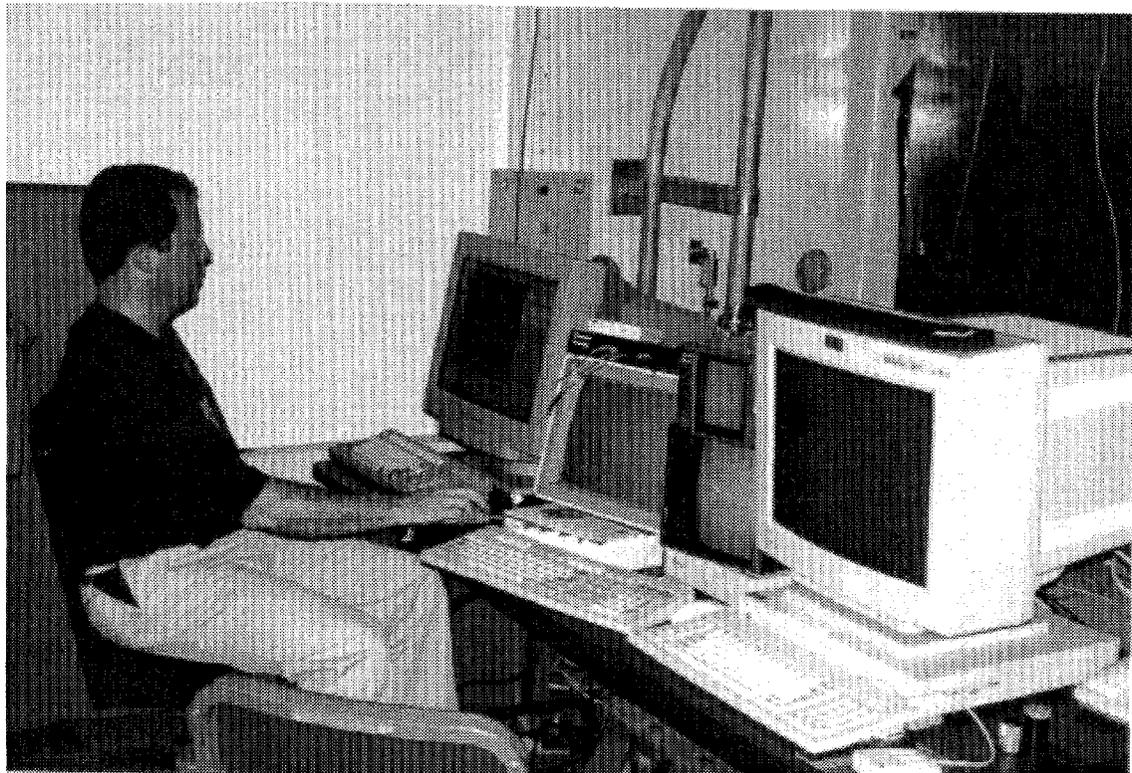


Fig. B.6. TMS control station.

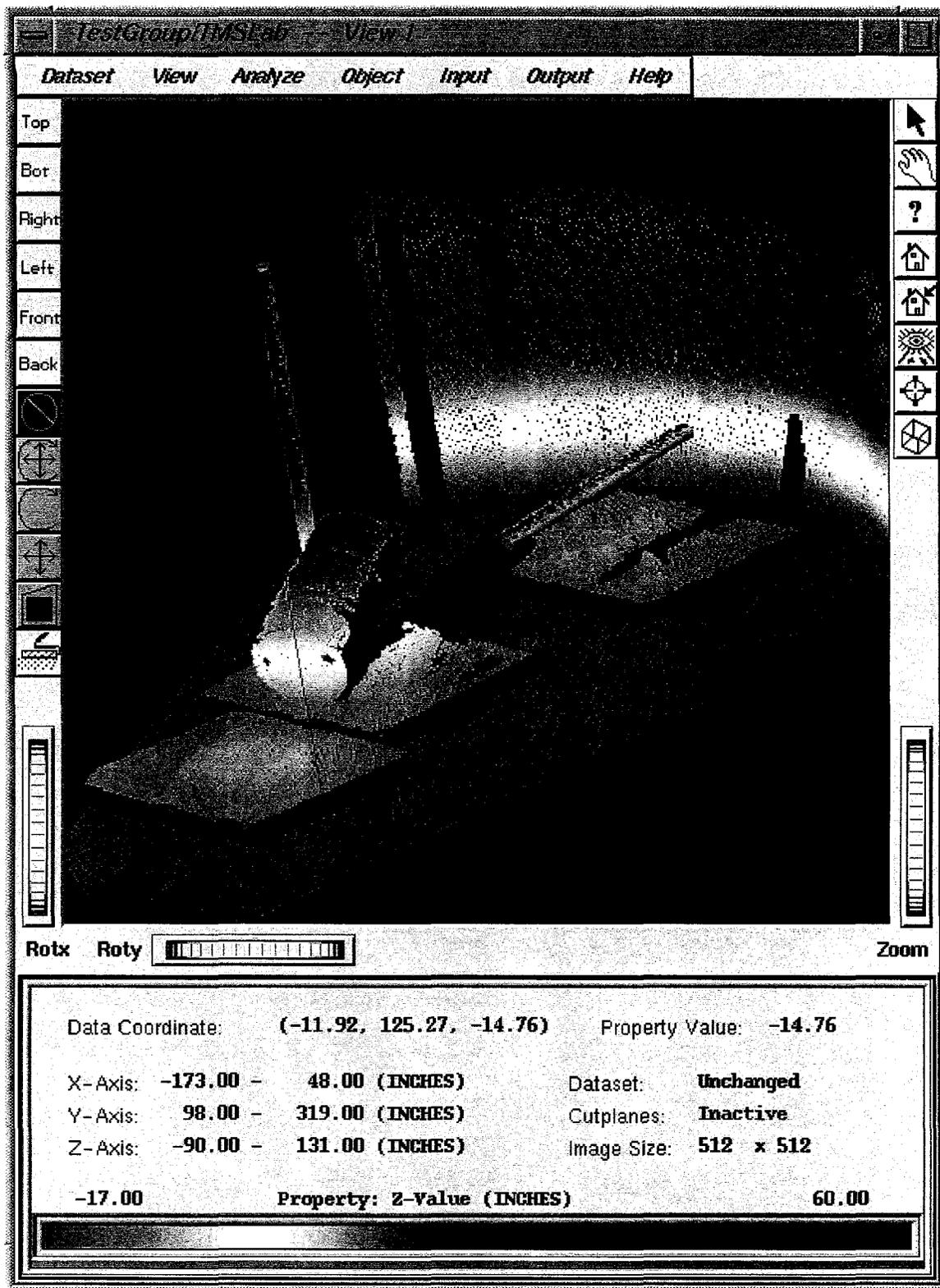
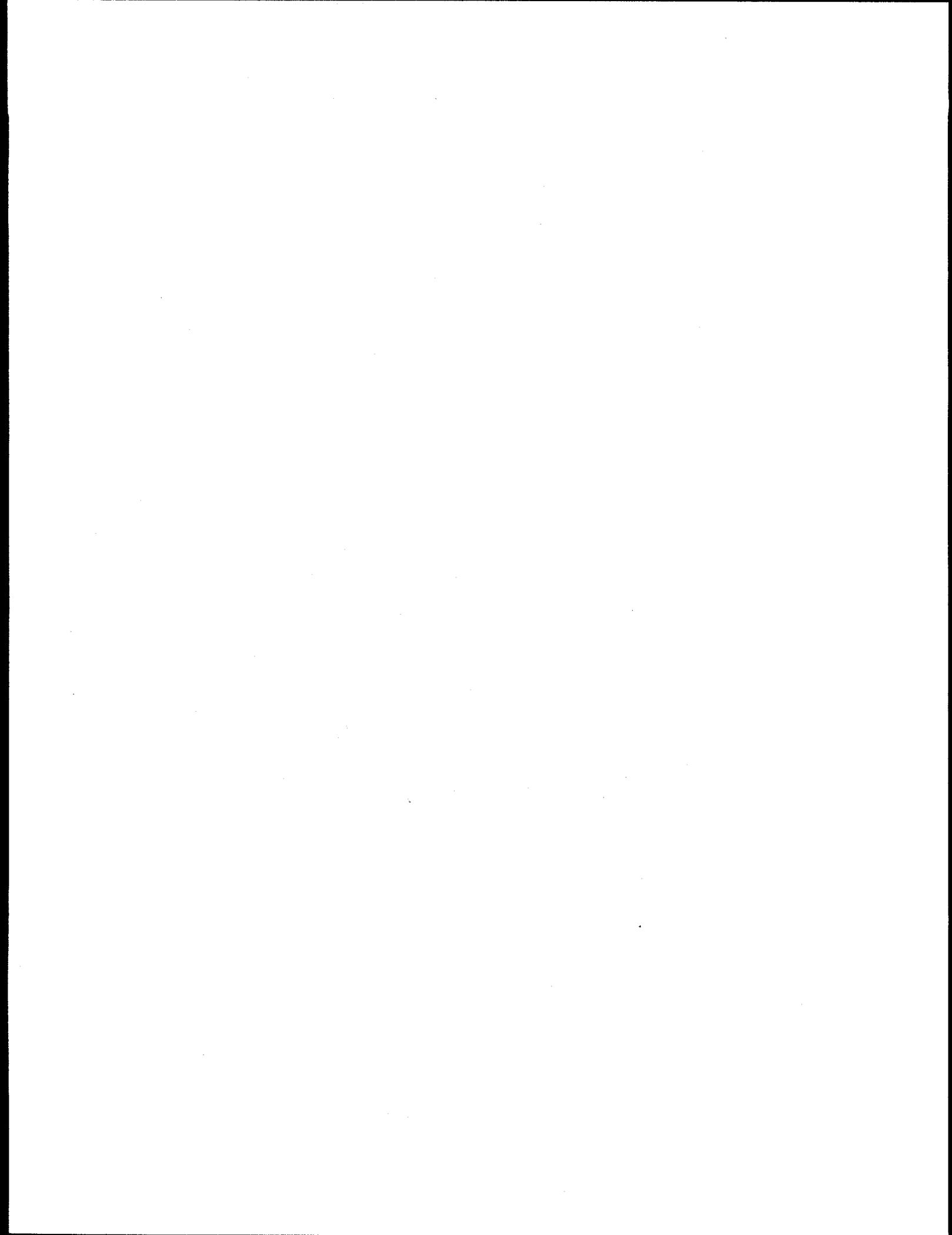


Fig. B.7. Surface map of MTT's Optics Lab, Albany, New York, displayed with the ICERVS.

APPENDIX C
TYPICAL UST DIMENSIONS



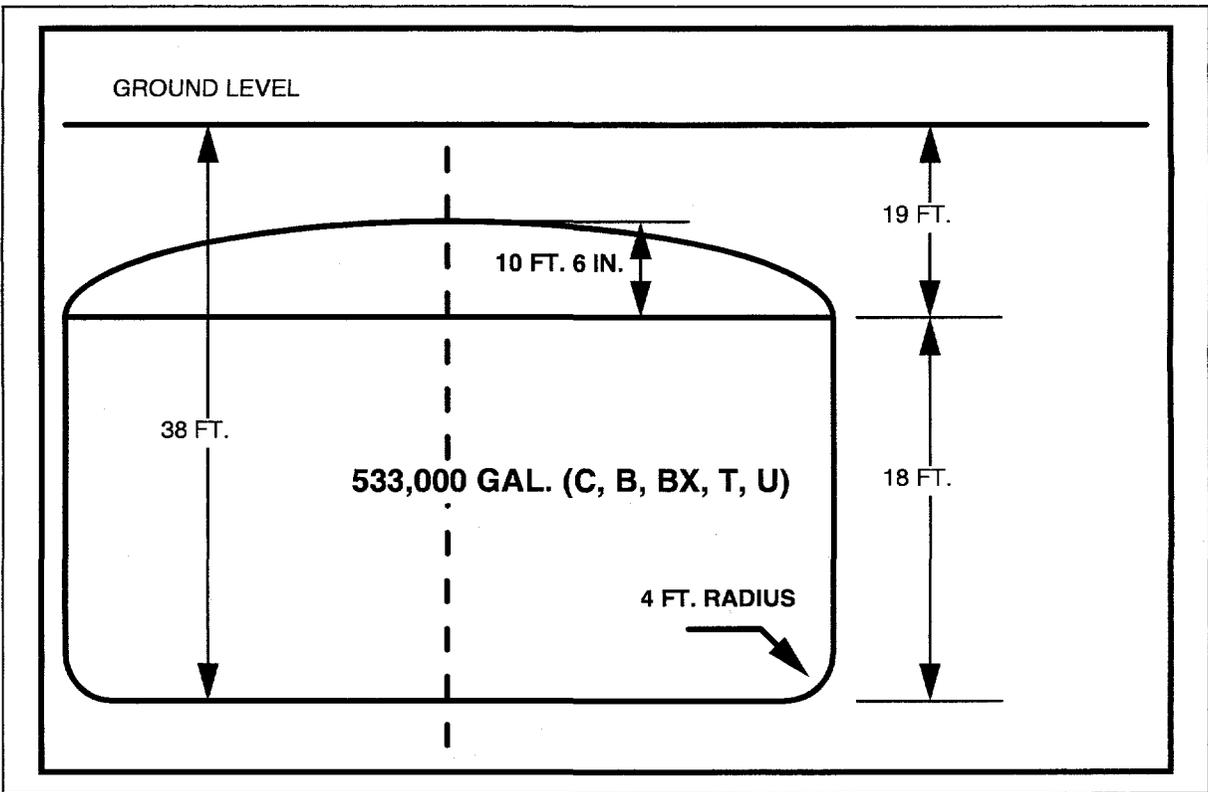


Fig. C.1. Typical configuration of a 75-ft type C, B, BX, T, and U single-shell tank.

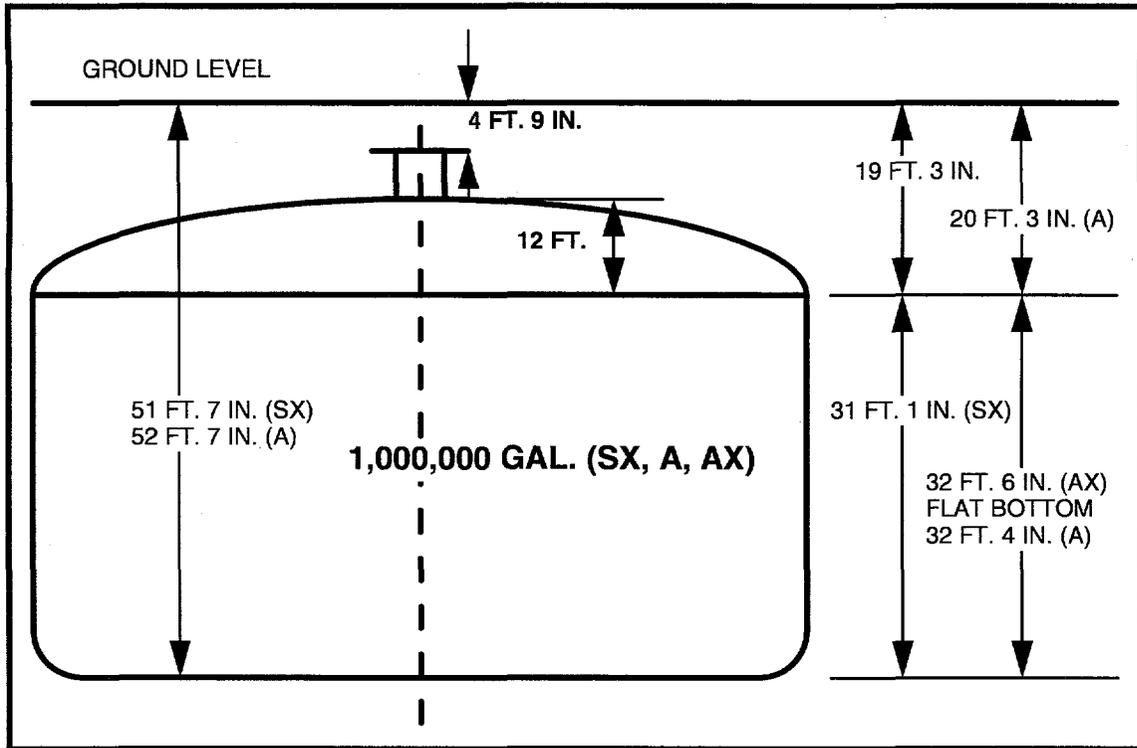


Fig. C.2. Typical configuration of a 75-ft type SX, A, and AX single-shell tank.

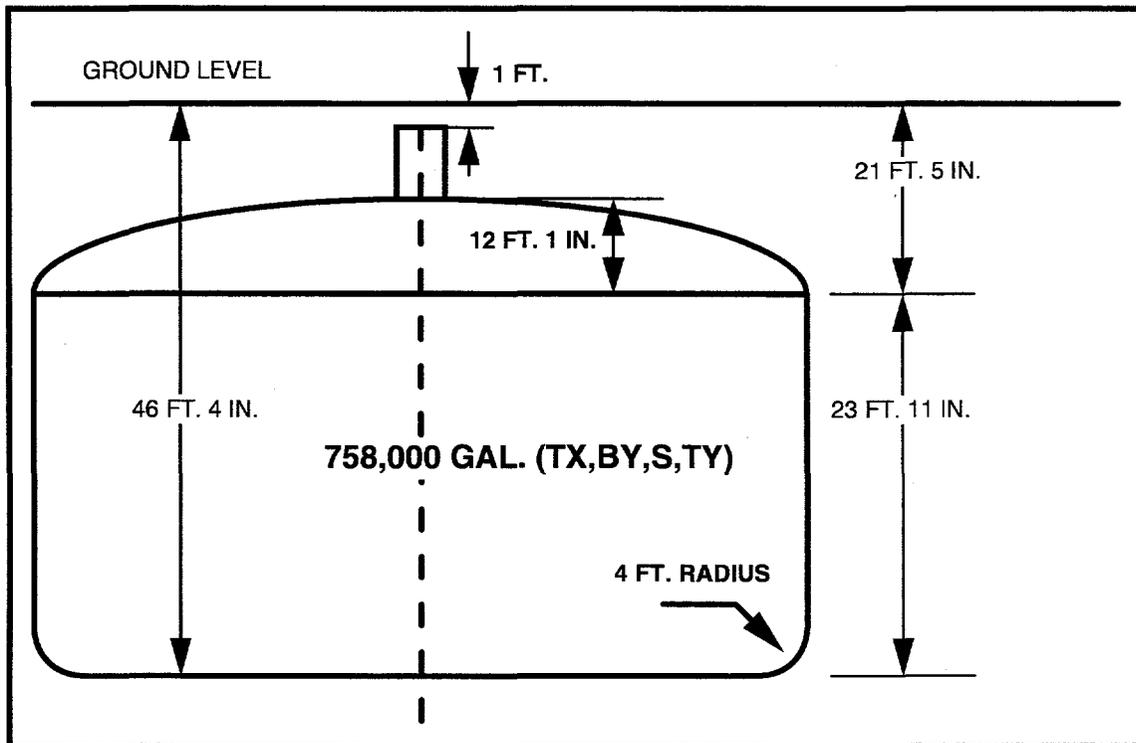


Fig. C.3. Typical configuration of a 75-ft type TX, BY, S, and TY single-shell tank.

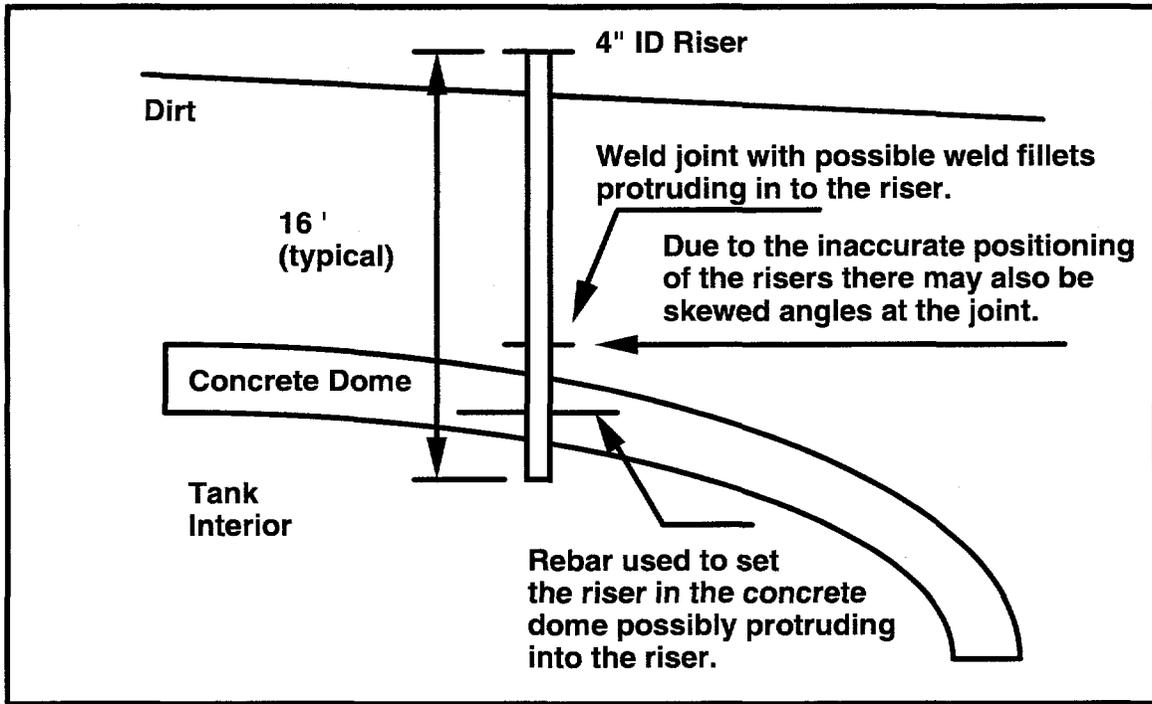
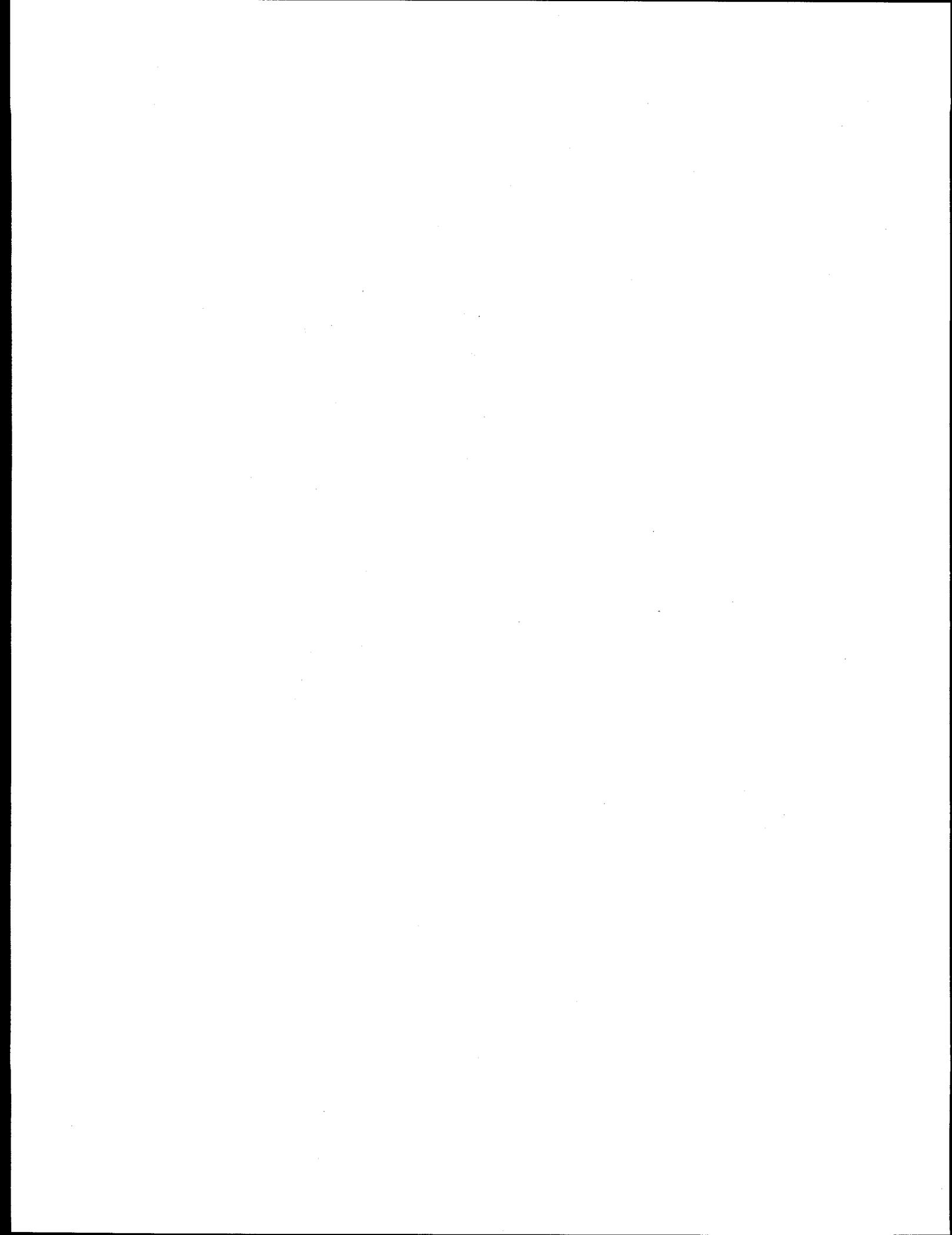
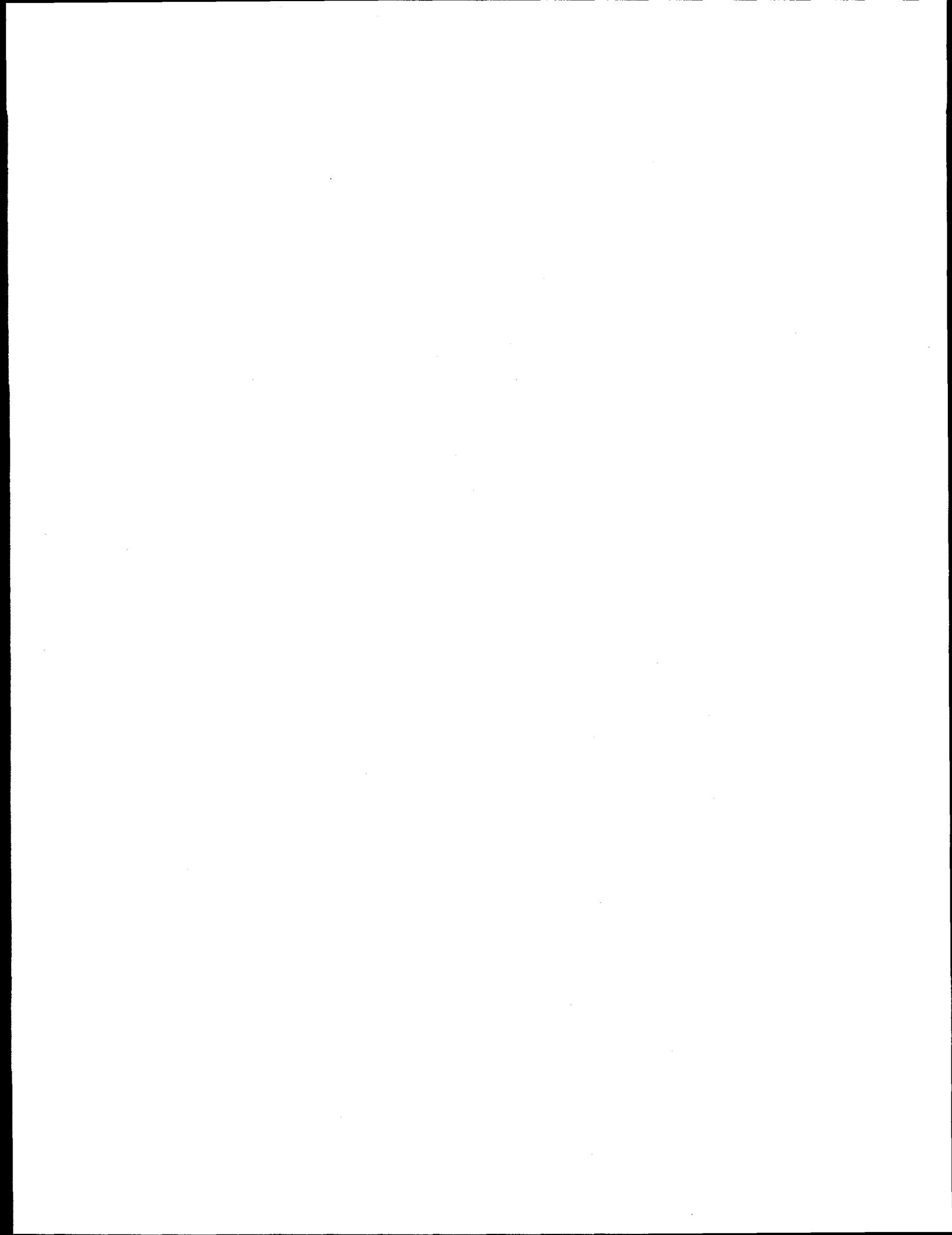


Fig. C.4. Schematic showing the access pipe from the surface to the tank shell.



APPENDIX D

VOLUMETRIC SOURCE CODE



```
/* -----
```

```
Filename:                vol.c  
Author:                  G. A. Armstrong  
Date:                    18 March 1997  
Description:              Calculate volume between two surface maps.
```

```
----- */
```

```
#include <stdio.h>  
#include <stdlib.h>  
#include <math.h>
```

```
#define TABLE  
/* #define SANDBOXLEFT */  
/* #define SANDBOXRIGHT */  
/* #define TEST */  
/* #define INCH_RES */  
#define TENTH_INCH_RES  
/* #define QUARTER_INCH_RES */  
/* #define HALF_INCH_RES */  
#define THRESHOLD 0.03
```

```
int check_limits ( int x, int y );
```

```
int add_maps ( float * mapptr1, float * mapptr2, float * mapptr3 );  
int sub_maps ( float * mapptr1, float * mapptr2, float * mapptr3, int * countptr1, int * countptr2 );  
int copy_map ( float * mapptr1, float * mapptr2 );  
int zero_map ( float * mapptr );  
int zero_count ( int * countptr );
```

```
int average_map ( float * mapptr1, float * mapptr2, int * countptr1 );  
float average ( float * mapptr, int x, int y );  
float interpolate_map ( float * mapptr1, float * mapptr2, int * countptr1 );  
float interpolate ( float * mapptr, int x, int y );  
float interpolatex ( float * mapptr, int x, int y );  
float interpolatey ( float * mapptr, int x, int y );  
float vol_calc( float * mapptr );  
void save_map ( float * mapptr, char* filename );  
void save_mtimap ( float * mapptr, char* filename );  
void read_map ( char* filename, float * mapptr, int * countptr );  
int avgbins ( float * mapptr, int * countptr );
```

```
#define TRUE 1  
#define FALSE 0
```

```
#ifdef TEST  
#define XCLIPMAX -20.0  
#define XCLIPMIN -60.0  
#define YCLIPMAX 300.00  
#define YCLIPMIN 250.00
```

```
#define ZCLIPMIN 0.00
#define ZCLIPMAX 10000.00
#endif
```

```
#ifdef TABLE
#define XCLIPMAX -20.0
#define XCLIPMIN -60.0
#define YCLIPMAX 300.00
#define YCLIPMIN 250.00
#define ZCLIPMIN 39.00
/* #define ZCLIPMIN 0.00 */
#define ZCLIPMAX 10000.00
#endif
```

```
#ifdef SANDBOXLEFT
#define XCLIPMAX -40.00
#define XCLIPMIN -85.00
#define YCLIPMAX 300.00
#define YCLIPMIN 215.00
#define ZCLIPMIN 9.00
#define ZCLIPMAX 20.00
#endif
```

```
#ifdef SANDBOXRIGHT
#define XCLIPMAX 30.00
#define XCLIPMIN -40.00
#define YCLIPMAX 300.00
#define YCLIPMIN 215.00
#define ZCLIPMIN 9.00
#define ZCLIPMAX 20.00
#endif
```

```
#ifdef TEST
#define XDIM 50
#define YDIM 70
#endif
```

```
#ifdef TABLE
#define XDIM 50
#define YDIM 70
#endif
```

```
#ifdef SANDBOXRIGHT
#define XDIM 100
#define YDIM 100
#endif
```

```
#ifdef SANDBOXLEFT
#define XDIM 100
#define YDIM 100
#endif
```

```
#ifdef INCH_RES
```

```

#define AREA (1.0*1.0)
#define MAP_SCALE 1
#endif

#ifdef HALF_INCH_RES
#define AREA (0.5*0.5)
#define MAP_SCALE 2
#endif

#ifdef QUARTER_INCH_RES
#define AREA (0.25*0.25)
#define MAP_SCALE 4
#endif

#ifdef TENTH_INCH_RES
#define AREA (0.1*0.1)
#define MAP_SCALE 10
#endif

#define XMAX (XDIM*MAP_SCALE)
#define YMAX (YDIM*MAP_SCALE)
#define XMIN 0
#define YMIN 0

float mapa[XMAX][YMAX];          /* mapa[X][Y] */
int counta[XMAX][YMAX];

float mapb[XMAX][YMAX];          /* mapb[X][Y] */
int countb[XMAX][YMAX];

float mapc[XMAX][YMAX];          /* mapc[X][Y] */

float mapd[XMAX][YMAX];          /* mapd[X][Y] */

float vol;

int x_index, y_index;

float x,y,z;
float dx,dy,dz;
float maxx, minx, maxy, miny, maxz, minz;
FILE *filestrm;

int n;
int nlines;
int status;

#define FAIL -1
#define OK 0

int no_avgs, no_suc_avgs, no_interps, no_suc_interps;

#define MAP(map,X,Y) *(map+((X)*(YMAX))+Y))

```

```

#define COUNT(count,X,Y) *(count+((X)*(YMAX))+Y))

float * mapptr;

/* ===== */
/* ----- */

Function:                main()
Author:                  G. A. Armstrong
Date:                    18 March 1997
Description:              Calculate volume between two surface maps.

----- */

main( int argc, char *argv[] )
{
no_avgs=0;
no_suc_avgs=0;
no_interps=0;
no_suc_interps=0;

printf(" Clipping Region \n");
printf(" X CLIP MAX = %f   X CLIP MIN = %f \n", XCLIPMAX, XCLIPMIN );
printf(" Y CLIP MAX = %f   Y CLIP MIN = %f \n", YCLIPMAX, YCLIPMIN );
printf(" Z CLIP MAX = %f   Z CLIP MIN = %f \n", ZCLIPMAX, ZCLIPMIN );

printf("\n\nARRAY DIMENSIONS XMAX = %d YMAX = %d \n", XMAX, YMAX );

zero_map ( &(mapa[0][0] ) );
zero_map ( &(mapb[0][0] ) );
zero_map ( &(mapc[0][0] ) );
zero_map ( &(mapd[0][0] ) );

zero_count ( &(counta[0][0] ) );
zero_count ( &(countb[0][0] ) );

status = OK;

minx = 10000000.0;
miny = 10000000.0;
minz = 10000000.0;

maxx = -10000000.0;
maxy = -10000000.0;
maxz = -10000000.0;

if ( argc > 1 )
{
    if ( (filestrm = fopen ( *++argv , "r" )) == NULL )
    {

```

```

printf("ERROR: Could not open filename = %s \n", *argv);
status = FAIL;
}
else
printf("NOTE: Finding MIN/MAX of MAPA from filename = %s \n", *argv);
}
else
{
printf("vol mapa mapb\n Output is volume of B-A in cubic inches");
status = FAIL;
}

n=0;
nolines=0;

if ( status == OK )
{
do {

n = fscanf (filestrm, "%f%f%f%f%f%f\n", &x, &y, &z, &dx, &dy, &dz );

if ( x >= XCLIPMIN && x < XCLIPMAX &&
y >= YCLIPMIN && y < YCLIPMAX &&
z >= ZCLIPMIN && z < ZCLIPMAX )
{
if ( n > 0 )
{
/* printf("%d: %f %f %f %f %f %f\n", n, x, y, z, dx, dy, dz); */

if ( x > maxx ) maxx = x;
if ( x < minx ) minx = x;

if ( y > maxy ) maxy = y;
if ( y < miny ) miny = y;

if ( z > maxz ) maxz = z;
if ( z < minz ) minz = z;

nolines++;
}
}
} while ( n > 0 );

fclose(filestrm);

if ( nolines )
{
printf(" No of points = %d \n", nolines);
printf(" XMAX = %f XMIN = %f \n", maxx, minx );
printf(" YMAX = %f YMIN = %f \n", maxy, miny );
printf(" ZMAX = %f ZMIN = %f \n", maxz, minz );
}
}

```

```

}

} /* end if status == OK */

/* ===== */

if ( argc > 2 )
{
    read_map ( *argv, &(mapa[0][0]), &(counta[0][0]) );
}
else
{
    printf("vol bottom_filename1 top_filename2 \n");
    status = FAIL;
    exit(-1);
}

/* ===== */

if ( argc > 2 )
{
    read_map ( *++argv, &(mapb[0][0]), &(countb[0][0]) );
}
else
{
    printf("vol bottom_filename1 top_filename2 \n");
    status = FAIL;
    exit(-1);
}

/* ===== */

printf("NOTE: Average the BINS in MAP A \n");

avgbins( &(mapa[0][0]), &(counta[0][0]) );

printf("NOTE: Average the BINS in MAP B \n");

avgbins( &(mapb[0][0]), &(countb[0][0]) );

/* ===== */

printf("NOTE: Interpolate the missing data points in MAP A \n");

zero_map ( &(mapd[0][0]) );
interpolate_map ( &(mapa[0][0]), &(mapd[0][0]), &(counta[0][0]) );

```

```
/* ===== */
```

```
printf("NOTE: Interpolate the missing data points in MAP B \n");
```

```
zero_map ( &(mapd[0][0] ) );  
interpolate_map ( &(mapb[0][0]), &(mapd[0][0]), &(countb[0][0]) );
```

```
/* ===== */
```

```
printf("NOTE: Interpolate the missing data points in MAP A \n");
```

```
zero_map ( &(mapd[0][0] ) );  
interpolate_map ( &(mapa[0][0]), &(mapd[0][0]), &(counta[0][0]) );
```

```
/* ===== */
```

```
printf("NOTE: Interpolate the missing data points in MAP B \n");
```

```
zero_map ( &(mapd[0][0] ) );  
interpolate_map ( &(mapb[0][0]), &(mapd[0][0]), &(countb[0][0]) );
```

```
/* ===== */
```

```
printf("NOTE: Average data points in MAP A. \n");
```

```
zero_map ( &(mapd[0][0] ) );  
average_map ( &(mapa[0][0]), &(mapd[0][0]), &(counta[0][0]) );
```

```
/* ===== */
```

```
printf("NOTE: Average data points in MAP B. \n");
```

```
zero_map ( &(mapd[0][0] ) );  
average_map ( &(mapb[0][0]), &(mapd[0][0]), &(countb[0][0]) );
```

```
/* ===== */
```

```
printf("NOTE: Subtract the Maps: MAP C = B - A (if they have non-zero counts). \n");
```

```
zero_map ( &(mapc[0][0] ) );  
sub_maps ( &(mapb[0][0]), &(mapa[0][0]), &(mapc[0][0]) , &(counta[0][0]), &(countb[0][0]) );
```

```
/* ===== */
```

```
printf("NOTE: Threshold the data and calculate the volume based on MAP C. \n");
```

```
vol = vol_calc ( &(mapc[0][0]) );
```

```

printf( "NOTE: Volume B-A = %f cubic inches \n", vol );

/* ===== */

#ifdef GAA
printf("NOTE: Interpolate the missing data points in MAP C \n");

zero_map ( &(mapd[0][0] ) );
interpolate_map ( &(mapc[0][0]), &(mapd[0][0]), NULL );

/* ===== */

printf("NOTE: Interpolate the missing data points in MAP C \n");

zero_map ( &(mapd[0][0] ) );
interpolate_map ( &(mapc[0][0]), &(mapd[0][0]), NULL );

/* ===== */

printf("NOTE: Threshold the data and calculate the volume based on MAP C. \n");

vol = vol_calc( &(mapc[0][0]) );

printf( "NOTE: Volume B-A = %f cubic inches \n", vol );

/* ===== */

printf("NOTE: Average data points in MAP C. \n");

zero_map ( &(mapd[0][0] ) );
average_map ( &(mapc[0][0]), &(mapd[0][0]), NULL );

/* ===== */

printf("NOTE: Threshold the data and calculate the volume based on MAP C. \n");

vol = vol_calc( &(mapc[0][0]) );

printf( "NOTE: Volume of C=B-A = %f cubic inches \n", vol );
#endif
/* ===== */

printf("NOTE: Save new map A in filename mapdiffa. \n");

save_map ( &(mapa[0][0]), "mapdiffa");

printf("NOTE: Save new map B in filename mapdiffb. \n");

```

```

save_map ( &(mapb[0][0]), "mapdiffb");

printf("NOTE: Save new map C in filename mapdiffc. \n");

save_map ( &(mapc[0][0]), "mapdiffc");

printf("NOTE: Save new map C in filename mapdiffc.mti for input to fusion program.\n");

save_mtimap ( &(mapc[0][0]), "mapdiffc.mti");

/* ===== */

```

```

} /* end of main() */

```

```

/* -----

```

```

Function:  check_max()
Author:   G. A. Armstrong
Date:    18 April 1997
Description:  Check the present values of X and Y against
              the limits.

```

```

Mods:

```

```

----- */

```

```

int check_limits ( int x, int y )
{
int status = OK;

    if ( x >= XMAX || x < XMIN || y >= YMAX || y < YMIN )
        status = FAIL;

return (status);
}

```

```

/* -----

```

```

Function:  save_mtimap()
Author:   G. A. Armstrong
Date:    18 March 1997
Description:  Save the data in a MTI map format. This is
              the format necessary for input into the
              data fusion algorithm written by Ole Dorum.

```

```

Mods:

```

```

----- */

```

```

void save_mtimap ( float * mapptr, char* filename )
{
static FILE * filestrm;
static int x_index, y_index;
static float z;
static int nprint;

nprint =0;

filestrm = fopen( filename,"w");

fprintf( filestrm, "\n\n*#d 03/25/97 10:00:00 PST 5 \n");

for (y_index=0; y_index<YMAX; y_index++)
{
for ( x_index=0; x_index<XMAX; x_index++ )
{
if ( (z=MAP(mapptr,x_index,y_index)) > THRESHOLD )
{
fprintf( filestrm, "%f %f %f \n",
(((float) (x_index))/MAP_SCALE)+minx,
(((float) (y_index))/MAP_SCALE)+miny,
z );
nprint++;
}
}

if ( nprint ) fprintf( filestrm, "\n\n*#d 03/25/97 10:00:00 PST 5 \n");

}

fclose( filestrm );

return;
}

```

/* ----- */

Function: save_map()
Author: G. A. Armstrong
Date: 18 March 1997
Description: Calculate the volume.

Mods:

----- */

```

void save_map ( float * mapptr, char* filename )
{
static FILE * filestrm;
static int x_index, y_index;
static float z;

filestrm = fopen( filename,"w");

```

```

for (x_index=0; x_index<XMAX; x_index++)
{
    for (y_index=0; y_index<YMAX; y_index++)
    {
        if ( (z=MAP(mapptr,x_index,y_index)) > THRESHOLD )
            fprintf( filestrm, "%f %f %f 0.000 0.000 0.000 \n",
                (((float) (x_index))/MAP_SCALE)+minx,
                (((float) (y_index))/MAP_SCALE)+miny,
                z );
    }
}

fclose( filestrm );

return;
}

```

/* -----

Function: read_map()
 Author: G. A. Armstrong
 Date: 18 March 1997
 Description: Calculate the volume.

Mods:

----- */

```

void read_map ( char* filename, float * mapptr, int * countptr )
{
    static FILE * filestrm;
    static int x_index, y_index;
    static n, nolines;

    if ( (filestrm = fopen ( filename , "r" )) == NULL )
    {
        printf("ERROR: Could not open filename = %s \n", filename);
        status = FAIL;
    }
    else
        printf("NOTE: Reading MAP from filename %s \n", filename);

    n=0;
    nolines=0;

    if ( status == OK )
    {
        do {

```

```

n = fscanf (filestrm, "%f%f%f%f%f%f\n", &x,&y,&z,&dx,&dy,&dz);

if (n > 0)
{
    nlines++;

/* printf("%s %d: %f %f %f\n", filename, n, x, y, z); */

if ( x >= XCLIPMIN && x < XCLIPMAX &&
    y >= YCLIPMIN && y < YCLIPMAX &&
    z >= ZCLIPMIN && z < ZCLIPMAX )
{
    x_index = abs( (int) (MAP_SCALE*(x - minx)) );
    y_index = abs( (int) (MAP_SCALE*(y - miny)) );

if ( x_index >= 0 && x_index < XMAX && y_index >= 0 && y_index < YMAX )
{
    MAP( mapptr, x_index, y_index) += z;
    COUNT( countptr, x_index, y_index) += 1;
}
else
{
    printf("WARNING: Found Index to array out of bounds in MAP from %s? x= %d y=%d\n",
        filename, x_index, y_index);
}

}
} /* if n > 0 */
else
{
    printf("NOTE: %s had %d lines or data points\n", filename, nlines );
}

} while ( n > 0 );

fclose(filestrm);

} /* end if status == OK */

return;
}

```

```

/* -----
Function:    copy_map ()
Author:     G. A. Armstrong
Date:      18 March 1997
Description: Copy map1 to map2.

```

Mods:

```
----- */  
  
int copy_map ( float * mapptr1, float * mapptr2 )  
{  
static int x_index;  
static int y_index ;  
int status = OK;  
  
for (x_index=XMIN; x_index<XMAX; x_index++)  
{  
    for ( y_index=YMIN; y_index<YMAX; y_index++ )  
    {  
        MAP(mapptr2,x_index,y_index) = MAP(mapptr1,x_index,y_index) ;  
    }  
}  
  
return(status);  
}  
  
/* -----
```

Function: zero_count ()
Author: G. A. Armstrong
Date: 18 March 1997
Description: Zero the map.

Mods:

```
----- */  
  
int zero_count ( int * countptr )  
{  
static int x_index;  
static int y_index ;  
int status = OK;  
  
for (x_index=XMIN; x_index<XMAX; x_index++)  
{  
    for ( y_index=YMIN; y_index<YMAX; y_index++ )  
    {  
        COUNT(countptr,x_index,y_index) = 0 ;  
    }  
}  
  
return(status);  
}  
  
/* -----
```

Function: zero_map ()
Author: G. A. Armstrong
Date: 18 March 1997

Description: Zero the map.

Mods:

```
----- */  
  
int zero_map ( float * mapptr )  
{  
static int x_index;  
static int y_index ;  
int status = OK;  
  
for (x_index=XMIN; x_index<XMAX; x_index++)  
{  
    for ( y_index=YMIN; y_index<YMAX; y_index++)  
    {  
        MAP(mapptr,x_index,y_index) = 0.0000 ;  
    }  
}  
  
return(status);  
}
```

```
/*-----
```

Function: sub_maps()
Author: G. A. Armstrong
Date: 18 March 1997
Description: Subtract the data in the first two maps and store in the third.

MAP3 = MAP1 - MAP2

Mods:

```
----- */  
  
int sub_maps ( float * mapptr1,  
              float * mapptr2,  
              float * mapptr3,  
              int * countptr1,  
              int * countptr2 )  
{  
static int x_index;  
static int y_index ;  
static count;  
int status = OK;  
  
for (x_index=XMIN; x_index<XMAX; x_index++)  
{  
    for ( y_index=YMIN; y_index<YMAX; y_index++)  
    {  
        if ( ((COUNT(countptr1, x_index, y_index)) != 0) &&  
            ((COUNT(countptr2, x_index, y_index)) != 0) )  
        {
```

```

        MAP(mapptr3,x_index,y_index) =
            MAP(mapptr1,x_index,y_index) - MAP(mapptr2,x_index,y_index);
    }
}

```

```

return(status);
}

```

```

/* -----

```

Function: add_maps()
 Author: G. A. Armstrong
 Date: 18 March 1997
 Description: Add the data in the first two maps and store in the third.

MAP3 = MAP1 + MAP2

Mods:

```

----- */

```

```

int add_maps ( float * mapptr1, float * mapptr2, float * mapptr3 )
{
    static int x_index;
    static int y_index ;
    static count;
    int status = OK;

    for (x_index=XMIN; x_index<XMAX; x_index++)
    {
        for ( y_index=YMIN; y_index<YMAX; y_index++ )
        {
            MAP(mapptr3,x_index,y_index) =
                MAP(mapptr1,x_index,y_index) + MAP(mapptr2,x_index,y_index);
        }
    }

    return(status);
}
/* -----

```

Function: avgbins()
 Author: G. A. Armstrong
 Date: 18 March 1997
 Description: Average the data in each bin.

Mods:

```

----- */

```

```

int avgbins ( float * mapptr, int * countptr )
{
    static int x_index;

```

```

static int y_index ;
static count;
int status = OK;

for (x_index=XMIN; x_index<XMAX; x_index++)
{
    for ( y_index=YMIN; y_index<YMAX; y_index++ )
    {
        if( count=COUNT(countptr,x_index,y_index) )
            MAP(mapptr,x_index,y_index) /= count;
    }
}

return(status);
}
/* -----

```

Function: vol_calc()
Author: G. A. Armstrong
Date: 18 March 1997
Description: Calculate the volume.

Mods:

```

----- */

float vol_calc ( float * mapptr )
{
    static float height;
    static float vol;

    static int x_index;
    static int y_index ;

    x_index =0;
    y_index =0;
    vol =0.0;
    height = 0.0;

    for (x_index=XMIN; x_index<XMAX; x_index++)
    {
        for ( y_index=YMIN; y_index<YMAX; y_index++ )
        {
            height = MAP(mapptr, x_index, y_index);
            if ( height > THRESHOLD ) vol += height*(AREA);
        }
    }

    return (vol);
}

/* -----

```

Function: average_map()
Author: G. A. Armstrong
Date: 18 March 1997
Description: Average the data in th map.

WARNING: DONOT ! run this with occlusions in the map file. FIRST,
cleanup the occlusions!

Mods:

```
----- */  
  
int average_map ( float * mapptr1, float * mapptr2, int * countptr1 )  
{  
static int x_index;  
static int y_index;  
static int nzeroes;  
  
int status = OK;  
  
nzeroes =0;  
  
zero_map ( mapptr2 );  
  
for (x_index=(XMIN+2); x_index<(XMAX-2); x_index++)  
{  
for (y_index=(YMIN+2); y_index<(YMAX-2); y_index++)  
{  
MAP(mapptr2, x_index, y_index) = average( mapptr1, x_index, y_index );  
if ( countptr1 != NULL )  
if ( MAP(mapptr2,x_index,y_index) != 0.00 && COUNT(countptr1,x_index,y_index) == 0 )  
{  
COUNT(countptr1,x_index,y_index) += 1;  
nzeroes ++;  
}  
  
} /* end for y_index */  
} /* end for x_index */  
  
copy_map ( mapptr2, mapptr1 );  
  
printf("NOTE: Number of zeroes filled in by averaging = %d \n", nzeroes );  
  
return (status);  
}
```

```
/* -----
```

Function: average()
Author: G. A. Armstrong
Date: 18 March 1997
Description: Sweep a 3x3 convotution mask over the map.

WARNING: DONOT ! run this with occlusions in the map file. FIRST,

cleanup the occlusions!

Mods:

```
----- */

float average ( float * mapptr, int x, int y )
{
static int count =0;
static float sum = 0.0;
static float result = 0.0;

count=0;
sum =0.0;
result = 0.0;

if ( ( (x+1) < XMAX) && ( (x-1) >= XMIN ) && ( (y+1) < YMAX) && ( (y-1) >= YMIN ) )
{
    sum += MAP( mapptr, (x), (y) );
    sum += MAP( mapptr, (x-1), (y+1) );
    sum += MAP( mapptr, (x-1), (y-1) );
    sum += MAP( mapptr, (x), (y+1) );
    sum += MAP( mapptr, (x), (y-1) );
    sum += MAP( mapptr, (x+1), (y+1) );
    sum += MAP( mapptr, (x+1), (y-1) );
    sum += MAP( mapptr, (x+1), (y) );
    sum += MAP( mapptr, (x-1), (y) );

    result = sum / 9.00;
}

return ( result );
}

```

```
/* ----- */
```

Function: Interpolate_map
Author: G. A. Armstrong
Date: 18 March 1997
Description: Fill in the missing data points in the mapc array
by linear interpolation.

Mods:

```
----- */

float interpolate_map ( float * mapptr1, float * mapptr2, int * countptr1 )
{
static int no_interps;
static int no_suc_interps;
static int x_index;
static int y_index;
static float z;

```

```

no_interps =0;
no_suc_interps =0;

zero_map ( mapptr2 );

for ( y_index=(YMIN); y_index<(YMAX); y_index++ )
{
    for ( x_index=(XMIN); x_index<(XMAX); x_index++ )
    {
        if ( (MAP(mapptr1, x_index, y_index )) == 0.00 )
        {
            no_interps++;
            z = interpolate ( mapptr1, x_index, y_index );
            if ( z != 0.00 )
            {
                no_suc_interps++;
                MAP(mapptr2, x_index, y_index ) =z;
                if ( countptr1 != NULL )
                    COUNT(countptr1, x_index, y_index ) +=1;
            }
        }
    }
}

add_maps ( mapptr1, mapptr2, mapptr1 );

printf("NOTE: No of Interpolates = %d No of Success = %d \n", no_interps, no_suc_interps);
}

```

/* ----- */

Function: Interpolate
Author: G. A. Armstrong
Date: 18 March 1997
Description: Fill in the missing data points in the mapc array
by linear interpolation.
Mods:

----- */

```

float interpolate ( float * mapptr, int x, int y )
{
    static float result =0.0;
    static float resultx =0.0;
    static float resulty =0.0;

    resultx = interpolatex ( mapptr, x,y);
    resulty = interpolatey ( mapptr, x,y);

    if ( (resultx != 0.0) && (resulty != 0.0) )
    {

```

```

    if ( fabs(resultx) > fabs(resulty) ) result = resultx;
    else result = resulty;
}
else if ( resultx != 0.0 ) result = resultx;
else if ( resulty != 0.0 ) result = resulty;
else result = 0.0;

return(result);

}
/* -----
Function:   Interpatex
Author:    G. A. Armstrong
Date:      18 March 1997
Description: Fill in the missing data points in the mapc array
            by linear interpolation.
Mods:
----- */

```

```

float interpolatex ( float * mapptr, int x, int y )
{
static int x_save, x_left, x_right, xx_index;
static int y_save, y_left, y_right, yy_index;
static float z_left, z_right, delta_z, z;
static int stop;
static float result;

result = 0.0;
z_left = 0.0;
z_right = 0.0;
delta_z = 0.0;

x_save = x;
y_save = y;
xx_index = x;
yy_index = y;
x_right = 0;
x_left = 0;

for ( xx_index = x_save; xx_index < XMAX; xx_index++)
{
    if ( (z=MAP( mapptr, xx_index, yy_index )) != 0.00 )
    {
        x_right = xx_index;
        z_right = z;
        break;
    }
}

for ( xx_index = x_save; xx_index >= XMIN; xx_index--)
{

```

```

        if ( (z=MAP( mapptr, xx_index, yy_index )) != 0.00 )
        {
            x_left = xx_index;
            z_left = z;
            break;
        }
    }

    if ( (z_left != 0.0) && (z_right != 0.0) )
    {
        delta_z = ( z_left - z_right ) / ((float)( x_left - x_right ));
        result = z_right + ((float)(x_save - x_right))*delta_z;
    }

    return( result );

} /* end interpolatex function */

```

/* -----

Function: Interpolatey
 Author: G. A. Armstrong
 Date: 18 March 1997
 Description: Fill in the missing data points in the mapc array
 by linear interpolation.
 Mods:

----- */

```

float interpolatey ( float * mapptr, int x, int y )
{
    static int x_save, x_left, x_right, xx_index;
    static int y_save, y_left, y_right, yy_index;
    static float z_left, z_right, delta_z;
    static int stop;
    static float result;

    result = 0.0;
    z_left = 0.0;
    z_right = 0.0;
    delta_z = 0.0;

    x_save = x;
    y_save = y;
    xx_index = x;
    yy_index = y;
    x_right = 0;
    x_left = 0;

    for ( yy_index = y_save; yy_index < YMAX; yy_index++)

```

```

{
    if ( (z=MAP(mapptr, xx_index, yy_index)) != 0.00 )
    {
        y_right = yy_index;
        z_right = z;
        break;
    }
}

for ( yy_index = y_save; yy_index >= YMIN; yy_index--)
{
    if ( (z=MAP(mapptr, xx_index, yy_index)) != 0.00 )
    {
        y_left = yy_index;
        z_left = z;
        break;
    }
}

if ( (z_left != 0.0) && (z_right != 0.0) )
{
    delta_z = ( z_left - z_right ) / ((float)( y_left - y_right ));
    result = z_right + ((float)(y_save - y_right))*delta_z;
}

return( result );

} /* end interpolatey function */

```

/* ===== END OF FILE ===== */

INTERNAL DISTRIBUTION

- | | |
|-----------------------|--------------------------------|
| 1-5. G. A. Armstrong | 30. R. C. Mann |
| 6. S. M. Babcock | 31. D. W. McDonald |
| 7. R. E. Barry | 32. C. S. Mims |
| 8. T. W. Burgess | 33. M. W. Noakes |
| 9-18. B. L. Burks | 34. K. E. Plummer |
| 19. K. E. Fricke | 35. B. S. Richardson |
| 20. M. J. Haire | 36. S. L. Schrock |
| 21. O. W. Hale | 37. P. T. Spampinato |
| 22. D. C. Haley | 38-42. H. Toy |
| 23. W. R. Hamel | 43. S. D. Van Hoesen |
| 24. J. N. Herndon | 44. H. R. Yook |
| 25. L. Holder | 45. Central Research Library |
| 26. M. A. Johnson | 46. Laboratory Records—RC |
| 27. D. L. Kington | 47-48. Laboratory Records—OSTI |
| 28. R. L. Kress | |
| 29. A. P. Malinauskas | |

EXTERNAL DISTRIBUTION

49. Steve Bardsley, Mechanical Technology, Inc., 968 Albany-Shaker Road, Latham, NY 12110
50. Clinton Bastin, Manager, LMR Reprocessing Projects, Division of Fuels and Reprocessing, Office of Facilities, Fuel Cycle, and Test Programs, NE-471, Department of Energy, Washington, DC 20545
51. D. W. Bennett, Pacific Northwest National Lab, P.O. Box 999, Richland, WA 99352
52. L. F. Blankner, Fusion and Nuclear Technology Branch, Energy Programs Division, Department of Energy, P. O. Box 2008, Oak Ridge, TN 37831-6269
53. Betty Carteret, Pacific Northwest National Lab, Battelle Blvd., P.O. Box 999, MS-K5-22, Richland, WA 99352
- 54-58. Cal Christensen, LITCO, P.O. Box 1625, 2525 Freemont Avenue, Idaho Falls, ID 83403-3710
59. S. A. Couture, Lawrence Livermore Laboratories, P.O. Box 808, L-437, Livermore, CA 64551
60. J. Degregory, U.S. Department of Energy, 19901 Germantown Road, Germantown, MD 20874-1290
61. Jeff Frey, Department of Energy-Hanford, EESB, RM 1462, MS-K8-50, Richland, WA 99352
62. Ron Gamache, Mechanical Technology, Inc., 968 Albany-Shaker Road, Latham, NY 12110

63. Pete Gibbons, Numatec-Hanford Company, P.O. Box 1300, MS-H5-61, Richland, WA 99352-1300
64. R. W. Harrigan, Sandia National Laboratories, P.O. Box 5800, Division 1414, Albuquerque, NM 87185
65. Brack Hazen, FERMC0, 26 Merchant Street, Cincinnati, OH 45246
66. D. L. Herman, Fernald Management Company of Ohio, P.O. Box 398704, Cincinnati, OH 45239-8704
67. R. M. Hollen, Los Alamos National Laboratory, P.O. Box 1663, MS-J-580, Los Alamos, NM 87545
68. V. J. Kothari, Federal Energy Technology Center, 3610 Collins Ferry Road, Morgantown, WV 26507-4579
69. Jim Lee, Battelle Labs, ETB, 3200 Q Street, Richland, WA 99352
70. J. R. Noble-Dial, 55 Jefferson EW 91, Oak Ridge, TN 37830
71. Al Noonan, Pacific Northwest National Lab, Battel Blvd., P.O. Box 999, MS-K9-91, Richland, WA 99352
- 72-76. Al Pardini, Pacific Northwest National Lab, Battel Blvd., P.O. Box 999, MS-K5-25, Richland, WA 99352
77. Jane Powel, 55 Jefferson EW 91, Oak Ridge, TN 37830
- 78-82. Fred R. Reich, SESC, P.O. Box 840, 1200 Jadwin MSB4-51, Richland, WA 99352
83. Todd Samuel, Pacific Northwest National Lab, Battelle Blvd., P.O. Box 999, MS-K5-22, Richland, WA 99352
84. Terri Stewart, Pacific Northwest National Lab, Battelle Blvd., P.O. Box 999, MS-K5-25, Richland, WA 99352
- 85-89. John Wagner, Mechanical Technology, Inc., 968 Albany-Shaker Road, Latham, NY 12110
- 90-94. Tom Walter, Mechanical Technology, Inc., 968 Albany-Shaker Road, Latham, NY 12110
95. C. R. Ward, Westinghouse Savannah River Company, Building 773-A, D-1145, Aiken, SC 29808
96. B. Wilding, Westinghouse Idaho Nuclear Company, P.O. Box 4000, Idaho Falls, ID 83403-5104
97. A. P. Williams, Westinghouse Hanford Company, P.O. Box 1970, LO-18, Richland, WA 99352
98. L. W. Yarbrough, Department of Energy, Albuquerque Operations Office P.O. Box 5400, Albuquerque, NM 87815-5400