NSSDC NATIONAL SPACE SCIENCE DATA CENTER

WORLD DATA CENTER A for ROCKETS AND SATELLITES 89-26

CORRELATIVE VISUALIZATION TECHNIQUES FOR MULTIDIMENSIONAL DATA

Lloyd A. Treinish
Craig Goettsche

National Space Science Data Center

October 1989

NASA
National Aeronautics and Space Administration

Goddard Space Flight Center

NSSDC/WDC-A-R&S 89-26

CORRELATIVE VISUALIZATION TECHNIQUES FOR MULTIDIMENSIONAL DATA

Lloyd A. Treinish
Craig Goettsche

National Space Science Data Center

October 1989

Correlative Visualization Techniques for Multidimensional Data

Abstract

Critical to the understanding of data is the ability to provide pictorial or visual representations of those data, particularly in support of correlative data analysis. Despite the advancement of visualization techniques for scientific data over the last several years, there are still significant problems in bringing today's hardware and software technology into the hands of the typical scientist. For example, there are other computer science domains outside of computer graphics that are required to make visualization effective such as data management. Well-defined, flexible mechanisms for data access and management must be combined with rendering algorithms, data transformations, etc. to form a generic visualization pipeline. A generalized approach to data visualization is critical for the correlative analysis of distinct, complex, multidimensional data sets in the space and Earth sciences. Different classes of data representation techniques must be used within such a framework, which can range from simple, static two- and three-dimensional line plots to animation, surface rendering and volumetric imaging. Static examples of actual data analyses will illustrate the importance of an effective pipeline in a data visualization system.

INTRODUCTION

The importance of data visualization, in which visualization is the method of computing that gives visual form to complex data utilizing graphics and imaging technology, has been recognized for some time. Only recently, however, has it begun to grow in importance among the scientific community in general. This concept has been particularly relevant for analyzing large volumes of complex (e.g., multidimensional) data streams that are available today from such sources as spacecraft instruments and supercomputer-based models and simulations. The human visual system has an enormous capacity for receiving and interpreting data efficiently. Hence, the processing power of the eye-brain system should be an intimate part of any effort to comprehend data (McCormick et al., 1987). Unfortunately, such data are often generated today without adequate consideration of the difficulty of their effective interpretation (i.e., extracting useful knowledge from data). This problem will be compounded by the next generation of data sources, which will literally bury the scientific community in bits. For example, NASA's Earth Observing System which is planned for deployment in the late 1990s, will have to receive, process and store one to ten TB (10^{12} to 10^{13} bytes) of data per day.

Despite advancement in data generation and computer technology over the last few decades, methods of managing and analyzing large volumes of complex data streams basically have not changed. Potentially, this situation leaves significant fractions of data not fully understood or scientific information undiscovered. Even with the availability of visualization technology as one very valuable technique in the study and analysis of large volumes of data, the importance of the organization, structure and management of such data and associated information about data or metadata must be stressed. Without such considerations, a user would be unable to take advantage of powerful visualization tools for arbitrary data of interest -- *to see the unseen*. The advent of powerful, relational data base management systems (RDBMS), which have become commercially viable over the last few years, only begins to address these problems. Unfortunately, RDBMS technologies generally have been practical only for metadata management in large, scientific applications (Campbell et al., 1989, and Treinish, 1988).

Nevertheless, a mechanism is still required for organizing the actual data, which may be complex, large in volume and resident on magnetic disk. Such data can be referenced by a RDBMS, but data management capabilities are still required at

the applications level for the actual data. Such a data (base) model must be matched to the structure and use of scientific data. One such mechanism is the Common Data Format (CDF). It is based upon the concept of providing data-independent, abstract support for a class of scientific data that can be described by a multidimensional block structure. It has been used to develop a number of generic data management, display and analysis tools for a wide variety of disciplines. Users of data-independent application systems, which are based upon CDF, rely on their own understanding of the science behind different sets of data to interpret the results, a critical feature for the multidisciplinary studies inherent in the space and Earth sciences (Treinish and Gough, 1987). CDF has become a standard method for storing data in these disciplines for a variety of applications. This abstraction consists of a software package and a self-describing data structure. The term "data abstraction" implies that CDF isolates the details of the physical structure of a data set from a user of such data (Shaw, 1984). The programmer using such an abstraction needs to know only about the collection of CDF operations and the logical organization of the data of interest, not the details of CDF storage or the underlying software structure. Therefore, CDF easily accommodates scientific data structures at the applications-programmer level rather than at the physical data level.

EFFECTIVE VISUALIZATION

A researcher's employment of tools to visualize data is an important mechanism in the data analysis process. In view of this, how does a scientist-user access data of interest and the appropriate visualization tools in a reasonable, acceptable manner? Most of the visualization software technology available today is *not* in a form that permits straightforward application to data of interest without significant assistance from experts. There are a plethora of graphics and imaging toolkits available from a variety of sources. Some of these toolkits are standards while others are proprietary (recent packages, such as PHIGS+, RenderMan, Wavefront, CubeTool, and Doré or older examples, like DISSPLA, Template, SAS, and MOVIE.BYU, a few of which are surveyed by King, 1987). However, as powerful as these software packages may be, they are, unfortunately, just boxes of tools, often only at a subroutine library level. The result is that such software is either unavailable to or unusable by the average scientist without significant graphics expertise. These software toolkits are typically not "turnkey" in nature,

have no mechanism for handling discipline or domain-driven problems or data, lack a standard or intuitive interface above a programming language level, and so forth. Most so-called visualization systems do not deal directly with science problems but rather are oriented to graphics and animation (i.e., points, vectors, polygons, bitmaps, voxels, lighting models, animation, etc. are addressed rather than multispectral images, geographic grids, electromagnetic tensor fields, atmospheric sounding, time histories, etc.) (Treinish et al., 1989, DeFanti, 1988 and McCormick et al., 1987).

Given the demands of modern research, scientists who are not computer-oriented rarely have the time or the interest to learn graphics protocols and standards, data structures, peculiarities of specific devices, rendering algorithms, etc., all of which are often required to use typical "visualization" toolkits. They can witness the substantial achievement represented by the class of scientific visualizations created by computer hardware and software vendors, supercomputer centers and government research laboratories. However, such examples are often customized for specific applications, involve considerable intensive work by visualization specialists, and are essentially unavailable to the typical scientist in most disciplines, beyond the acquisition of a copy on video tape. If, as computer graphicists claim, visualization can be truly effective, if not revolutionary, for use by scientists in their routine research, the technology must be interactive and operate in the scientist's terms, not in those of an arcane collection of software tools.

Science Requirements

The support of correlative data analysis (i.e., working with data from a variety of sources to study a problem of scientific interest) requires an obvious focus on generic visualization via the development of discipline-independent visualization techniques. This implies the ability to examine many different parameters from disparate data sets in the same fashion for visual correlation, a function well-suited to the capabilities of the human visual system. However, there must also be many common visualization schemes so that any one set of parameters may be studied through different mechanisms since not all visualization techniques show all aspects of data. Such visualization functionality must be available at a "high-level", with a consistent user interface enabling the scientist to easily access the full capability of the software.

Therefore, discipline-independent visualization implies the development of software that handles arbitrary data sets and possesses different tools for displaying data. In other words, data management is as important a component of a data visualization system as underlying graphics and imaging technology. To implement a system that provides these features in a practical fashion, the management of and access to the data must be decoupled from the actual visualization software. Within such a system, there must be a clean interface between the data and the display of the data so that arbitrary data can be accessed by the visualization software. In other words, an appropriate data model is needed that accommodates the access and structure of scientific data on one hand, and the requirements of visualization software on the other. CDF is one example of such a data model. In addition, a common intuitive user interface for the selection of techniques for data presentation and manipulation is required. As a consequence of such an approach, a software system of this design has an open framework. It can ingest arbitrary data objects for visualization, and other visualization techniques can be added independent of the application. These abilities imply a significant reduction in long-term software development costs because new data sets do not require new display software, and new display techniques do not require new data access software.

The NSSDC (National Space Science Data Center) Graphics System (NGS), for example, provides an interactive, discipline-independent toolbox for non-programmers to support the visualization of data. In order to utilize the NGS, data of interest must be stored in terms of the aforementioned Common Data Format (CDF). The NGS supports the ability to display or visualize any arbitrary, multidimensional subset of any data set by providing a large variety of different representation schemes, all of which are supported by implicit animation (i.e., slicing of a data set into sequences). Treinish, 1989, discusses the basic design, interface, implementation and applications of the NGS. So as not to duplicate that material, the following sections elaborate on the underlying architecture of visualization software, and how visualization techniques should be used to support correlative data analysis. These discussions do apply to the NGS but are not covered by Treinish, 1989.

VISUALIZATION PIPELINE

Generic data visualization implies the geometric or visual representation of arbitrary, multidimensional data from a variety of sources or disciplines. Given well-defined, flexible mechanisms for data access and management, there exist specific rendering algorithms, data transformations, etc. that can be cast into a generic framework, when part of a visualization pipeline. A generalized approach to data visualization is very valuable for the correlative analysis of distinct, complex, multidimensional data sets in the space and Earth sciences. Different classes of data representation techniques are required since they each may show different aspects of data. Such techniques may range from simple, static two- and three-dimensional line plots to animation, surface rendering and volumetric imaging. Only some of these techniques may be relevant for specific data sets. Hence, a wide variety of representations schemes are necessary to accommodate a disparate collection of data. The key is a basic structure -- *the visualization pipeline* -- for generic visualization, which permits the scientist-user to control the flow of data under study and promotes exploratory data analysis.

Figure 1 is a schematic for a visualization pipeline that meets the above requirements. The implementation of such a pipeline must be under a uniform interface to permit consistent user access to each portion thereof. The interface supports the ability to flow data through the pipeline interactively to foster visual analysis. It provides a virtual view of the pipeline. Such a structure permits an iterative invocation of each function on demand. Of course, for the pipeline to be effective, these functions must be accessible at a scientific level.

Data Selection

On the left side of *Figure 1*, the data management portion is shown schematically. It provides the capability to select data sets of interest. In an end-to-end data system this management kernel may vary from traditional data catalogs and inventories or a highly interactive, intelligent information system with imbedded semantics. In either case, tools are provided, at some level, to help a user understand and select appropriate data sets. For example, correlative data analysis may require the selection of several "parallel" data sets of potentially different structures (e.g., one or more CDFs). This data management capability is required as the data flow to the right in the diagram, in which the data are subset-

ted. From the chosen data sets a user will select only those variables or parameters of interest. Optionally, only a portion of the selected domain of the parameters may be required. Therefore, the capability to window or filter out the undesired section(s) of the data set(s) is needed.

Transformation

The next portion of the pipeline deals with manipulating the data to meet the requirements of the analysis. For example, if the data chosen are not in the desired form for display or do not match the required form for the rendering method, then the data must be reorganized. This implies operations such as rescaling, mapping to another coordinate system, converting the units and projecting the data geographically. If however, the data are already in the form required for display, then no transformation is required.

Gridding

If the rendering method is for continuous data and the data are not continuous, then a uniform grid must be created at some desired resolution. This requirement also arises from gridded data that have been transformed into a nonuniform grid or from data accessed from multiple sources with dissimilar grid resolutions. In either case, operations such as curve and surface fitting, meshing, krieging, interpolation, smoothing, scaling and averaging are required. However, if the data are already in a uniform grid at the desired resolution (e.g., a simple image) then the data may be rendered as they are.

Figures 2 and *3* illustrate examples of simple gridding schemes which are an important part of a visualization pipeline and are among the techniques available in the NGS. *Figure 2* shows the concept behind nearest neighbor gridding, in which the cells of a grid are populated by extracting values from the points in the original grid, which are spatially nearest. Such a technique is valuable because it preserves the original data values and distribution of a grid after a coordinate transformation may have taken place on a collection of points. In addition, it is computationally inexpensive.

Figure 3 shows the concept behind weighted average gridding, in which, for any given cell in a grid, the weighted average of the n nearest values in the original data distribution (grid or collection of points) spatially nearest to that cell has been chosen. The selection of points is done after any required coordinate transforma-

tion. A weighting factor, $w_i = f(d_i)$, where d_i is the distance between the cell and the *ith* $(i = 1,...,m)$ point in the original grid structure after any transformation, is applied to each of the n values. *Figure 3* illustrates the case where $n = 3$, which is utilized in the NGS along with $w = d^{-2}$. See Renka, 1988 and Shepard, 1968 for further discussions on this class of gridding algorithms.

Rendering

The result of any data selection or manipulation within this visualization pipeline is the actual data display. A user may choose one or more visualization primitives ("vps"), which are inherently mapped to the dimensionality of the data to be displayed. A "vp" is a member of a class of primitives based upon some geometry (e.g., vector, polygon, raster/pixel, volume/voxel). Within each geometric class exists a collection of "vps". For example, vector and polygon-based primitives can be divided into two subclasses: discrete (e.g., xy[z], two- and three-dimensional histograms, etc.) and continuous (e.g., two- and three-dimensional contours, three-dimensional surface meshes, two- and three-dimensional field flows, etc.). The implementation of each "vp" within the visualization pipeline can be decomposed into two portions: geometric modelling and the actual rendering. Such a decomposition maximizes the flexibility to operate on a variety of data streams. However, to be effective from the user perspective, the user must have some control over the presentation of data. Some of this authority may be as mundane as annotation, but it is critical for a posteriori interpretation of a visualization. The visualization pipeline, through its user interface, provides the ability to choose from a suite of "vps", while the underlying modelling and rendering software is hidden. Other presentation control factors may be specific to a single "vp" (e.g., the selection of the intervals between contour lines). The user must have the capability to assign any of the parameters in the data set to any of the (virtual) axes consistent with a specific "vp." In addition, animation as a sequencing of frames according to a variable that has been mapped to an "animation axis" is critical.

For example, the generic rendering of shaded surfaces from multidimensional data sets within this pipeline is particularly challenging. The decoupling of the rendering process from the prerequisite geometric modelling permits a user to choose a general geometric model of a three-dimensional surface (e.g., sphere, parallelpiped) for a data stream of interest. This is as simple as global topography represented as data streams of latitude, longitude and height above sea level,

which are then mapped onto a spherical model. Such surfaces can then be independently rendered as wire-frame or smooth-shaded surfaces on typical workstations with hardware support for three-dimensional graphics. In addition, static images of high quality shaded (Phong or ray-traced) surfaces can be generated via software renderers.

Options associated with this class of visualization primitives are the use of pseudo-color for overlays of a data stream and animation for sequences of such models as described above. The pseudo-color, which maps a continuous color spectrum to a scaled quantitative range of values, can represent the same data stream as the "third" dimension, and hence act as a depth-cueing device in viewing the model. Alternatively, the pseudo-color can represent a "fourth" dimension, which, in the case of the topographic data, can be, for example, temperature as a function of latitude and longitude. Animation sequences can show the changes in all data streams (i.e., up to four) from one model to the next with respect to some other variable within the selected data set(s) (e.g., time). Although such a progression of animation frames are independent of the original data structures and display hardware, current workstation technology with support for three-dimensional graphics can accept sequences derived from disparate data sets and display them in real time.

Support

The components of the visualization pipeline require additional support in operational environments through optional features. As discussed earlier, the user must have control over the presentation of visualization primitives operating on data. However, such influence must extend to that portion of the pipeline between data selection and rendering, data transformation and gridding. Under transformation, the key area is the ability to deal with different coordinate systems and to be able to transform data among them (e.g., cartesian and polar in two dimensions; and cartesian, spherical and cylindrical in three dimensions). An important subclass of such transformations pertains to geographic mapping. Obviously, an effective visualization system for Earth scientists must be capable of displaying data geographically through a variety of map projections (e.g., Mercator, stereographic, etc.). Such functionality, however, must be very flexible. A user must be able to select an appropriate projection (i.e., to flatten the earth) to preserve, for example, distance or shape; to choose an arbitrary geographic window to view the data with a map overlay of desired resolution; and to invoke such map-

ping with any visualization primitive. These characteristics of different map projections have been used by cartographers for centuries (Pearson, 1984). The ability to map any data to a particular geographic projection is also important for correlative analysis. By selecting the specific projection, the viewing vector and a magnification factor, a variety of data can be studied over the exact same geographic region within a consistent visual domain. Such generic geographic registration is not bound to any particular data stream. Since this class of coordinate system transformations really applies to any visualization of data on a spherical surface, the mapping functionality should also be available for data that are not located on the Earth. (Ni et al., 1988 contains a discussion of a geographic mapping package and applications of map projections.)

The second area of support applies to data manipulation in the gridding and transformation portions of the pipeline. The user must be able to select easily (and understand) the specific algorithms for reorganizing data flowing through the pipeline to choose different techniques to explore the implications of such data manipulations. These abilities include choices such as polynomial vs. cubic spline curve fitting, nearest neighbor selection vs. averaging for meshing, and linear vs. quadratic interpolation. A corollary to such an approach is the ability to handle point and continuous data in the same fashion. Hence, a user can grid a collection of points and render them as a continuous data set, or decompose a continuous data set into a collection of points and display them using a discrete visualization primitive. This method is very valuable for visual data correlation, for example, when trying to compare ground truth (point data) with spacecraft imagery (continuous data).

In the gridding as well as the rendering portions of the pipeline, a problem often occurs when specific techniques are applied to large data sets. Gridding algorithms, such as the ones discussed earlier, typically require searching through an entire data set. If the data set is large, the selection of a data subset that is also large via a simple linear search becomes prohibitively expensive. The computation time for such a search typically increases as n^2, where n is related to the number of points being examined. Very useful in alleviating this problem are hierarchical data structures, for which the overhead to generate the appropriate tree structure for the data set is justified when the number of data points is sufficiently large (Samet and Webber, 1988a and 1988b).

For example, the storage and sampling of large, three-dimensional (e.g., geographic) data sets are improved by placing the data in an oct-tree. Data values can be located by latitude and longitude within the oct-tree structure which supports fast retrieval. In addition, the oct-tree assists the calculation of data for any given (geographic) location, which is accomplished by rapidly examining the actual data near the specified location and deriving a value via a specific gridding or interpolation algorithm. This technique can be valuable for correlating data by providing a consistent geographic reference among data sets that are of different resolutions or are not geographically registered.

Oct-trees can also be used to build a spherical surface model as a geodesic sphere, for which further subdivision implies a more-refined model at greater computational expense. For the display of cell arrays (e.g., pseudo-color images via vector/polygon protocols), quadtree-based rectangular subdivision compresses the number of cells (i.e., polygons) that actually are transmitted to a device. This process reduces transmission and rendering time because contiguous areas of similar data classification are combined to form one graphics primitive. The NGS uses this technique to support displays of high-resolution pseudo-color imagery on simple color graphics devices accessible via low-bandwidth communications.

EXAMPLES

The aforementioned NSSDC Graphics System (NGS) represents a prototype implementation of a generic visualization pipeline. Its effectiveness has been demonstrated by its operational use to support scientific research in a number of disciplines. To illustrate the discussion of the pipeline in the previous section, a number of examples follow below. These examples have been generated by controlling the flow of data through the NGS's implementation of the pipeline and the presentation of the data through different visualization primitives. This collection of visualizations can serve as an assessment of uses of a visualization pipeline to correlate and study complex data. These figures were originally created in color but have been reproduced herein as black and white images. Hence, only some of their inherent qualities are apparent. Nevertheless, these examples do illustrate the importance of supporting many visualization techniques and the need for iterative display in the exploratory data analysis process. Such functionality can be critical for the study of observational data sets as opposed to ones generated by

simulation or computational models. Artifacts of the observation process (e.g., data gaps, orbital footprints, sensor resolution, instrument degradation, etc.) must be accommodated and compensated for in data visualization.

All of the examples have been generated on a DEC VAX 8650 under VMS at the NSSDC using a number of different graphics devices. *Figures 4* through *14* and *22* through *26* were generated by the use of a commercial graphics package, Template developed by TGS, Inc. of San Diego, which the NGS employs for the underlying graphics environment and device support. The other figures utilized hardware-specific and locally developed software. Treinish, 1989 discusses the implementation of the NGS and offers sample visualizations from a number of different disciplines.

A data set of current public and scientific interest is derived from the Total Ozone Mapping Spectrometer (TOMS) on board NASA's Nimbus-7 spacecraft. Observations from TOMS have been invaluable to scientists studying the global distribution of ozone. The key data set from TOMS is in the form of daily world grids and is archived at the NSSDC from late 1978 through the present. In fact, the entire data set resides on line in CDF because of its importance to the scientific community. These data have become increasingly valuable as they indicate the presence of the so-called ozone hole over the south pole, which is the result of the reduction in total ozone observed during the Antarctic spring in recent years. The total ozone content as derived from the TOMS observations is indicated in terms of Dobson Units in the subsequent figures. One hundred Dobson Units are equivalent to a one millimeter column of ozone at standard temperature and pressure. These gridded TOMS data are provided in their own unique nonuniform grid (37,440 cells per grid) over the Earth. For latitude the values are in degrees north (-90° to +90°). For longitude the grid values are in degrees east (-180° to +180°). All 37,440 cells in each of these grids imply a cell size of 1.0 degree in latitude, pole to pole. The nonuniformity of the grid in longitude is such that for latitudes (north and south) between the equator and 50° the longitude cell size is 1.25°; for latitudes between 50° and 70° the longitude cell size is 2.5°; and for latitudes above 70°, the longitude cell size is 5° (Fleig, et al., 1986a). Hence, the tools of the NGS visualization pipeline provide a mechanism for analyzing the data independent of the limitations of the specific structure of the original data.

For the purpose of this sample study, the analysis of ozone data from 1986 has been selected. *Figure 4* is a simple histogram derived from all of the points in the

gridded data set for 1986 (about 13.7 x 10^6 points). The NGS pipeline allows the TOMS data set to be treated as essentially dimensionless or flat for the preparation of statistical summaries like histograms. The number of times a total ozone value occurs in Dobson Units within the illustrated collection of total ozone bins is shown along with the mean and standard deviations as well as percentile levels.

To study the ozone hole in more detail, one can continue to use the left-hand portion of the visualization pipeline (as shown in *Figure 1*) to select a specific subset of the TOMS data set beyond simply selecting the year, 1986. Although the following examples are static and emphasize techniques primarily for the study of the spatial characteristics of the data, time can be used for one of the virtual axes in the pipeline, including animation. For further study, October 10, 1986 has been chosen, because of the depth of the Antarctic ozone hole on that particular day.

Figure 5 shows the basic daily TOMS grid as a simple cartesian plot, and illustrates the nonuniformity of the grid structure as discussed earlier. It is critical to gain an understanding of the structure of the data prior to attempting more sophisticated visualizations. *Figure 6* still deals with untransformed data for October 10, but these data now have been restructured. Although global mapping techniques are important for seeing the spatial distribution of data across some geographic field, they may not be sufficiently quantitative at a detailed level for some applications. In other words, it is often useful to treat continuous or gridded data as scattered points and vice versa. The pipeline is used to slice the ozone data into 10° latitude bands as an animation sequence of 18 frames. The data from the second frame of this sequence (i.e., -70° to -80° latitude) are plotted as a function of longitude. Specifically, the individual ozone cell values from that latitude band have been extracted from the world grid and are shown as a meridional distribution in this scatter diagram.

In order to focus on the ozone hole itself, transformation of the data must take place. In particular, several of the following examples concentrate on approximately the southern two-thirds of the southern hemisphere. *Figure 7* shows the entire global distribution of the ozone grid on a cartesian map (i.e., cylindrical equidistant projection). The scatter diagram is similar to the one in *Figure 5*. However, each point, as represented by a box, has been color coded according to level of total ozone and displayed on a map. This technique is useful to understand the actual (discrete) data distribution prior to further transformation or gridding that may alter that distribution. In addition to the actual grid structure,

other artifacts of the observation process are also apparent as gaps in the data set. *Figure 8* is the same representation as in *Figure 7*, but uses an azimuthal equidistant map projection showing the desired portion of the southern hemisphere. This map projection and geographic window now begin to illustrate the spatial structure of the ozone hole and the non-uniformity of the grid structure with respect to latitude is also visible.

Given an appropriate data transformation (i.e., the map projection), the data must be regridded according to that projection for use with a continuous visualization primitive. Different visualization primitives illustrate the characteristics of this transformed grid structure. In this case, the use of a continuous visualization primitive is required to impart spatial sense to the ozone hole structure. For these examples, a resolution of 30 x 30 cells has been chosen. A nearest neighbor algorithm, as described earlier, has been used to populate each cell of the grid with an ozone value. In other words, for any given cell in the 30 x 30 grid, the value for ozone in the original grid structure spatially nearest to that cell has been chosen, after being transformed to an azimuthal equidistant map projection. The 30 x 30 (transformed) grid is rectangular and uniform so that each cell is a square. *Figure 9* is a simple contour (iso-lines of total ozone in Dobson Units) of that transformed grid with a world map overlay using the same geographic window as in *Figure 8*. Since the viewing window is in fact not a square, the grid resolution has been extended in the horizontal direction to preserve the uniformity of the grid. Hence, there are still 30 cells in the vertical direction. *Figure 10* is a pseudo-color image of the same grid in the same window as in *Figure 9*. The pseudo-color spectrum (blue to red) has been mapped to a range of Dobson Unit values. Although the geometric modelling in these visualization primitives is simple (the geographically transformed grid), it has been decoupled from the actual rendering as outlined above, since the same model has been used for both the contour and the image display. For this specific geographic window, the visualization as an image gives more information about the ozone hole structure than the contour plot.

The same grid or model can also be used to define a three-dimensional wire mesh, in which the third dimension corresponds to the total ozone. Such an "extension" of the model is shown in *Figure 11*. *Figure 12* shows the surface as in *Figure 11* except shaded with the same pseudo-color spectrum used in *Figure 10*. Such a height mapping begins to dramatize the concept of a hole in the ozone

layer, while the color enhances this perception as terrain color would improve a topographic map. However, this technique loses some of the quantitative detail available from other visualization techniques, which do not visually convey the hole as effectively.

Given the different visual characteristics of each of the continuous primitives shown so far, different gridding and even transformations may be required for an optimal display from each primitive. *Figure 13* is a contour map as in *Figure 9*, but over a smaller area so that Antarctica fills the viewing window. Choosing the smaller geographic window eliminates the cluttering of contour lines and yields more effective quantitative information about the ozone distribution. The nominal grid used in this example consists of 60 x 60 cells. The contour lines have been incremented every 20 Dobson Units, and the statistics in this example are based upon the southern hemisphere data only. The yellow contour lines represent those values above one standard deviation above the mean; the blue lines represent those values below one standard deviation below the mean; and the red lines cite intermediate values. This type of display shows the ozone hole structure with greater numeric precision than is possible with the pseudo-color image illustrated in *Figure 14*. This image utilizes a nominal grid of 200 x 200 pixels and covers the same geographic region as do *Figures 9* and *10*. This presentation now shows the correlation between the ozone level and the coastlines. It should also be noted that the rendering time of the image was reduced by compressing the cell array via a quadtree as discussed earlier.

A weighted averaging algorithm, as described earlier, has been used to determine the ozone value for each cell in the grids used to generate *Figures 13* and *14*. In these cases, $n = 3$ nearest values for ozone has been chosen, after being transformed to an azimuthal equidistant map projection. A weighting factor, $w = d^{-2}$, has been computed after the data were transformed geographically. To reduce the computational time for the gridding, the data were projected into an oct-tree. Hierarchical search techniques sped up the time to search for the nearest three points to compute the weighted average for each cell in the final grid.

The technique used to generate the simple cartesian surfaces in *Figures 11* and *12* is a static one, which is limited to low-resolution meshes. In order to see more detail and view the data from different geographic perspectives, one must consider other rendering techniques, which can, for example, be enhanced in a framework operating on typical graphics devices with hardware support for

three-dimensional graphics. Such devices permit interactive geometric transformations of surface meshes. *Figures 15* through *20* were generated from this class of graphics equipment supported by the NGS, namely Megatek 9000-series of terminals via device-specific software. Since the rendering and modelling portions of the software are separate, a more modern and portable implementation of the rendering code has been recently completed using PHIGS+ on a Silicon Graphics Personal IRIS (4D/20).

Figure 15 shows a wire frame representation with pseudo-coloring of the total ozone over the entire earth on October 10, 1986. The object, which consists of a 137 x 137 cartesian mesh (18,769 polygons), has been rotated in real time to show the ozone hole region on the right. This mesh has also been generated via a weighted averaging algorithm. *Figure 16* incorporates Gouraud shading of the mesh according to the same pseudo-color scheme. The ozone hole and adjacent ridge regions are visible in some detail on the right-hand side of the object.

As is apparent from the cartesian displays in *Figures 15* and *16*, geographic data like TOMS ozone obviously are distorted when the inherently spherical geometry is ignored. Essentially there is loss of geographic coherence via such a visualization technique. For two-dimensional visualization primitives, cartographic techniques as illustrated above, solve the problem. However, in three dimensions a generic spherical geometry must be considered. For example, the ozone data can be modelled as a geodesic sphere by projecting the geographic data into an octtree. For instance, a sphere which is deformed from a smooth sphere with a nominal radius according to the amount of total ozone can be used with a number of different renderers. *Figures 17* through *21* are all rendered examples of such spherical models. In each case the radius and the color scale (blue to red) correspond to total ozone in Dobson Units from 140 to 540, as with many of the preceding illustrations. To improve the quality and resolution of the resultant image, each model, which consists of a number of spherical triangles, can be further subdivided (e.g., each polygon gives birth to a new generation of smaller, connected polygons) at a cost in computation time. Thus, a given generation of subdivision has four times the number of triangles as the previous generation and requires four times as much computation time.

Figure 17 shows the wire-frame mesh of the first generation (80 triangles) of the spherical geodesic. *Figure 18* simply shows the model with flat, pseudo-color shading. *Figures 19* and *20* show the results of subdividing four more generations

(20,480 triangles) in wire-frame and shaded renderings, respectively. In these cases, the objects have been rotated in real time to show the ozone hole. If the model is subdivided one more generation (81,920 triangles), it can no longer be accommodated by the aforementioned hardware renderer used to create *Figures 17* through *20*. In this case, a software ray-trace renderer has been used with the same geometric modeler, in which the casting of shadows helps to emphasize the spatial structure.

Figure 21 is an example of choosing such an expensive but high quality presentation from the visualization pipeline. The view is directly over the south pole with the prime meridian vertical. Other visualization techniques might emphasize only the detailed, quantitative nature of the spatial structure as shown earlier. This technique emphasizes both qualitative and quantitative information about the total ozone. Since the ozone hole is quite prominent in the center, this particular picture has been dubbed the "ozone asteroid". The image achieves a better balance between spatial coherence and quantitative detail not present in other visualizations of these data. The proper spatial relationship of the hole and ridge structure is illustrated (i.e., geographically as well as with respect to the data themselves -- the ozone high appears to only partially fill the ozone hole). The color mapping was chosen to support both pseudo-color spectrum scaling and shadowing simultaneously, and be accommodated on an eight-bit frame buffer. Therefore, four bits are used for color and four bits are used for intensity.

Figure 22 illustrates how the aforementioned ozone hole data from October 10, 1986 relate to the rest of that year. Temporal and spatial consideration are shown by the zonal (i.e., by latitude) distribution of total ozone as a pseudo-color image over all of 1986. The nominal grid used for this image consists of 180 x 180 cells (i.e., each cell in the y-direction corresponds to one degree in latitude). A nearest neighbor algorithm has been used to populate the latitude-time grid from the original volume of data. A summation of the evolution of the Antarctic ozone hole can easily be seen as the blue region near the bottom of the image, which begins to grow in September and dissipates in November. Portions of the image appear in black due to gaps in the observations while the TOMS instrument was not operating (e.g., periods of local darkness during polar winter).

Despite the relative simplicity of this sample study of the TOMS ozone, it illustrates the power of an effective visualization pipeline. However, to indicate the

value of such an approach for correlative data analysis, a few additional data sets are examined and then compared to the visual analysis of the TOMS data.

A major advantage of this class of techniques are, of course, in their ability to support correlative data visualization. Nimbus-7 possesses another instrument which measures ozone, called the Solar Backscattered UltraViolet (SBUV) spectrophotometer. The SBUV observations are at a much lower global resolution than the TOMS data, and are available for the same time period as the TOMS data from the NSSDC. Rather than global grids, one of the SBUV data sets is organized as atmospheric profiles, in which measurements are available at different levels in the atmosphere along discrete orbital tracks. These profiles can be integrated to yield total ozone values (Fleig, et al. 1986b). The same geometric modelling and subsequent rendering techniques provided by the visualization pipeline can provide immediate visual correlation of these data sets. The SBUV total ozone data from October 10, 1986 are shown in *Figure 23* over the Antarctic region with the data classification scheme employed in *Figures 7* and *8*, and using the same geographic window as in *Figure 13*. The individual tracks show the instrument's observation path via their respective orbital footprints. The discrete SBUV observations have been spatially integrated using the aforementioned weighted average oct-tree technique to yield a transformed, interpolated grid with a nominal 250 x 250 resolution. The grid has been rendered as the pseudo-color image in *Figure 24*. The result is a different visualization of the ozone hole than the one derived from the TOMS data. Artifacts of the gridding process are quite apparent, especially if the discrete data distribution is compared to the image. Despite that fact, the coarse geometric features in *Figure 24* do correspond with those in *Figures 13* and *14*. To compare the SBUV and TOMS ozone data on a microscopic level, the October 10, 1986 data from each data set are displayed together. *Figure 25* shows the meridional distribution of the polar data (-60° to -90° south latitude) by plotting TOMS total ozone (red squares) and SBUV total ozone (green circles) against common ozone and longitude scales. The key to the visual correlations between TOMS and SBUV data in *Figures 24* and *25* is the ability of the scientist to display the data on common temporal and spatial bases.

To show that this approach can work with data that are not observations of the Earth's ozone layer from NASA spacecraft, consider *Figure 26*, which is a pseudo-color image of the temperature of the Earth's surface using the same geographic window as in *Figure 14*. This map is derived from a data set developed by

the U. S. Navy Fleet Numerical Oceanography Center (FNOC) based upon 12-hour forecasts by the Navy's Operation Global Atmospheric Prediction System. FNOC has been accumulating the results of these numerical models, which includes many meteorological parameters, since the early 1960s. The scope of this data set has expanded in recent years to include, for example, global coverage every 12 hours since 1983 (FNOC, 1986). A nominal grid of 200 x 200 cells has been derived from the model-based global surface temperatures for October 10, 1986 at 00:00 GMT and displayed in *Figure 26*. No correlation between these data and the ozone data can be seen, which would be expected since the ozone values are derived from remotely sensed observations of the Earth's stratosphere.

By having access to easy-to-use tools to reorganize data, a scientist can easily scrutinize the data at many different levels through disparate techniques. A plethora of visualization primitives properly coupled with powerful manipulation functions promotes the (visual) exploration of data and thus, enables a scientist to extract knowledge from complex data.

ACKNOWLEDGEMENTS

The research reported in this paper has been supported mainly by NASA's Office of Space Science and Applications (OSSA). The authors thank the significant contributions to this work by former members of the NSSDC staff, Michael Gough, Kuo-Piu Ni and David Wildenhain. The authors also acknowledge James L. Green, Director of the NSSDC, for his continued support of the concept of discipline-independent software systems to support scientific research. In addition, the authors greatly appreciate the review of this manuscript by Robert Cromp.

REFERENCES

1. CAMPBELL, W.J., SHORT, N. M., JR, AND TREINISH, L. A. Adding intelligence to scientific data management. *Computers in Physics 3*, 3 (May 1989).

2. DEFANTI, T. A. Cultural roadblocks to visualization. *Computers in Science 2*, 1 (January 1988).

3. FLEET NUMERICAL OCEANOGRAPHY CENTER (FNOC). Meteorological and oceanographic models. *FNOC Numerical Environmental Products Manual II* (1986), Commanding Officer, FNOC, Monterey, CA 93943.

4. FLEIG, A. J., BHARTIA, P. K., WELLEMEYER, C. G., AND SILBERSTEIN, D. S. Seven years of total ozone from the TOMS instrument - a report on data quality. *Geophysical Research Letters 13*, 12 (November 1986a), 1355-1358. Much literature has been published on Nimbus-7 TOMS, which discusses the instrument and observations themselves, scientific interpretation and analysis of the TOMS data, and the morphology, structure and evolution of global ozone, including the Antarctic ozone hole. For more information on this literature and related data sets and studies, a few of which are referenced in this paper, contact the user support office of the NASA Climate Data System at the National Space Science Data Center via internet, NCDSUSO@nssdca.gsfc.nasa.gov; SPAN, NSSDCA::NCDSUSO; or telephone: 301-286-5033.

5. FLEIG, A. J., BHARTIA, P. K., AND SILBERSTEIN, D. S. An assessment of the long-term drift in SBUV total ozone data. *Geophysical Research Letters 13*, 12 (November 1986b), 1359-1362.

6. MCCORMICK, B. K., DEFANTI, T. A., AND BROWN, M. D. Visualization in scientific computing. *Computer Graphics 21*, 6, (November 1987).

7. KING, J. F. Graphics draw a new picture for science. *Computers in Physics 1*, 1 (November-December 1987), 50-57.

8. NI, K-P, GOUGH, M. L. AND TREINISH, L. A. A flexible, Template-based software package for generating maps. 88-05 (February 1988), NSSDC, NASA/GSFC, Greenbelt, MD. This document can be acquired by contacting the National Space Science Data Center via internet, request@nssdca.gsfc.nasa.gov; SPAN, NSSDCA::Request; or telephone: 301-286-6695.

9. PEARSON, F. II. Map Projection Methods. 1984, Sigma Scientific, Inc., Blacksburg, VA.

10. RENKA, R. J. Multivariate interpolation of large sets of scattered data. *ACM Transactions on Mathematical Software, 14*, 2 (June 1988), 139-148.

11. SAMET, H., AND WEBBER, R. E. Hierarchical structures and algorithms for computer graphics part I: fundamentals. *IEEE Computer Graphics and Applications 8*, 3 (May 1988a), 48-68.

12. SAMET, H., AND WEBBER, R. E. Hierarchical structures and algorithms for computer graphics part II: applications. *IEEE Computer Graphics and Applications 8*, 4 (July 1988b), 59-75.

13. SHAW, M. Abstraction techniques in modern programming languages. *IEEE Software 1*, 4 (October 1984), 10-26.

14. SHEPARD, D. A Two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 23rd National ACM Conference* (1968), 517-524.

15. TREINISH, L. A., AND GOUGH, M. L. A software package for the data-independent storage of multi-dimensional data. *Eos Transactions American Geophysical Union 68*, 28 (July 14, 1987), 633-635.

16. TREINISH, L. A. Discipline-independent data visualization. *Course #19: Visualization Techniques in the Physical Sciences, Fifteenth Conference of ACM SIGGRAPH* (August 1988), 119-207.

17. TREINISH, L. A. An interactive, discipline-independent data visualization system. *Computers in Physics 3*, 4 (July 1989).

18. TREINISH, L. A., HABER, R.B., FOLEY, J. D., CAMPBELL, W. J. AND GURWITZ, R. F. Effective software systems for scientific data visualization. *Panel Session at the Sixteenth Conference of ACM SIGGRAPH* (August 3, 1989).

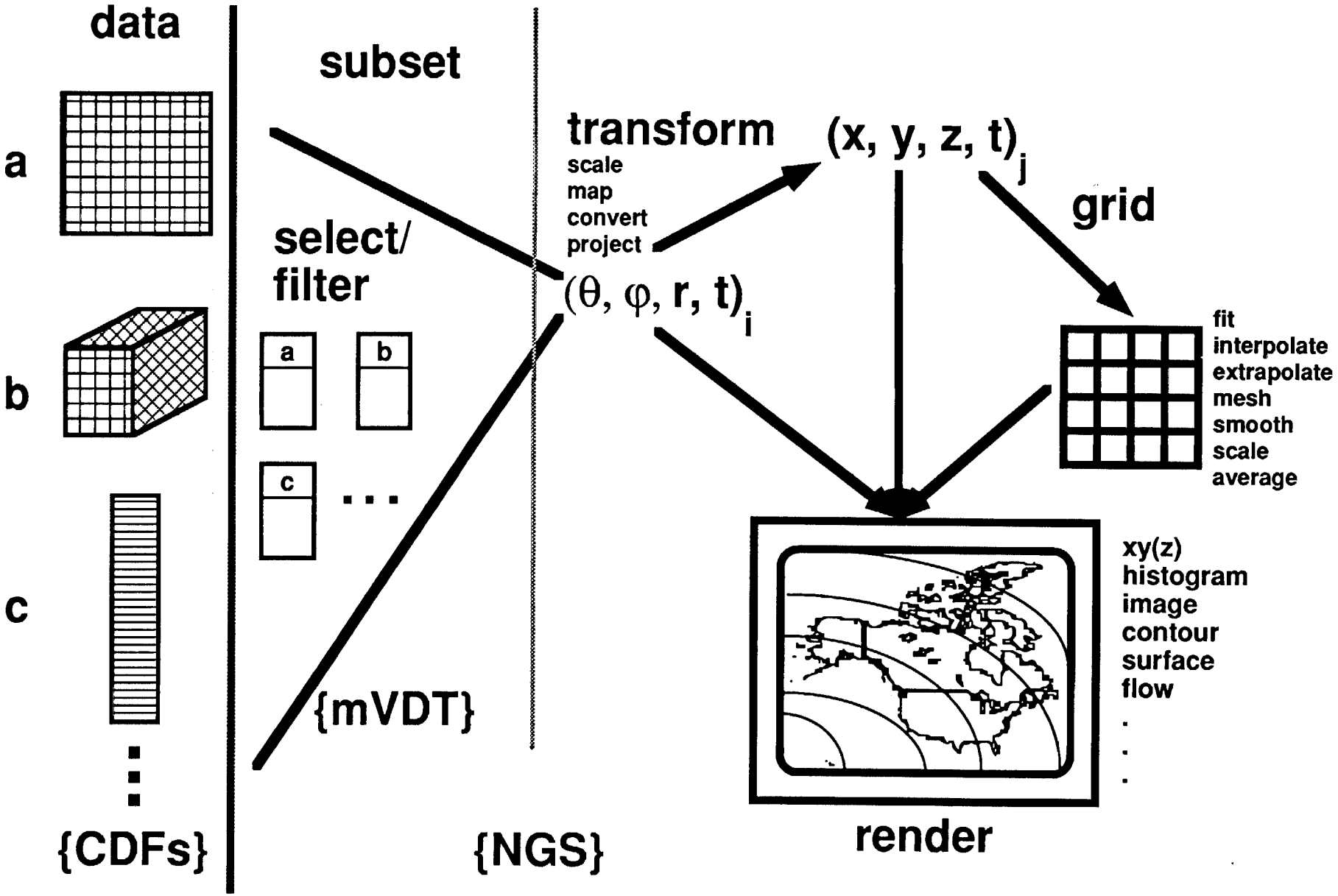


Figure 1. Effective, Generic Visualization Pipeline

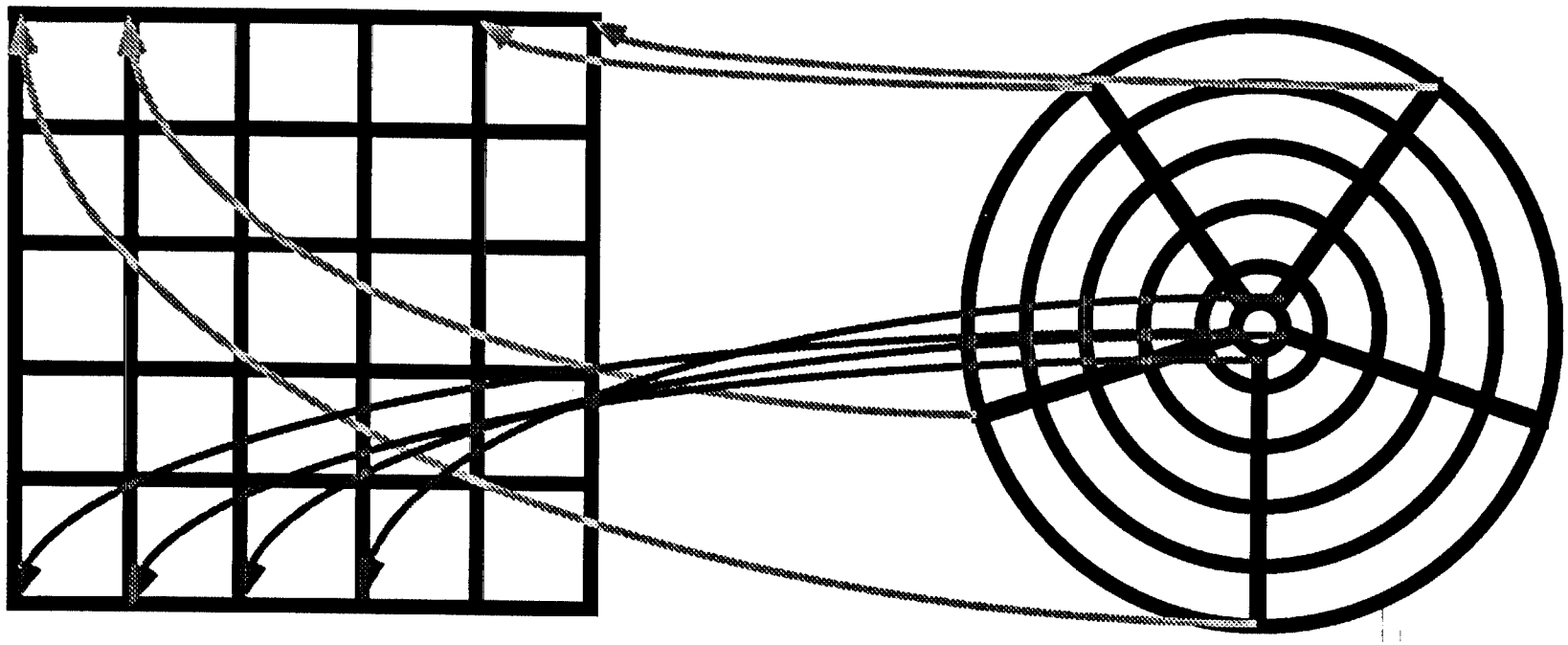


Figure 2. Nearest Neighbor Gridding

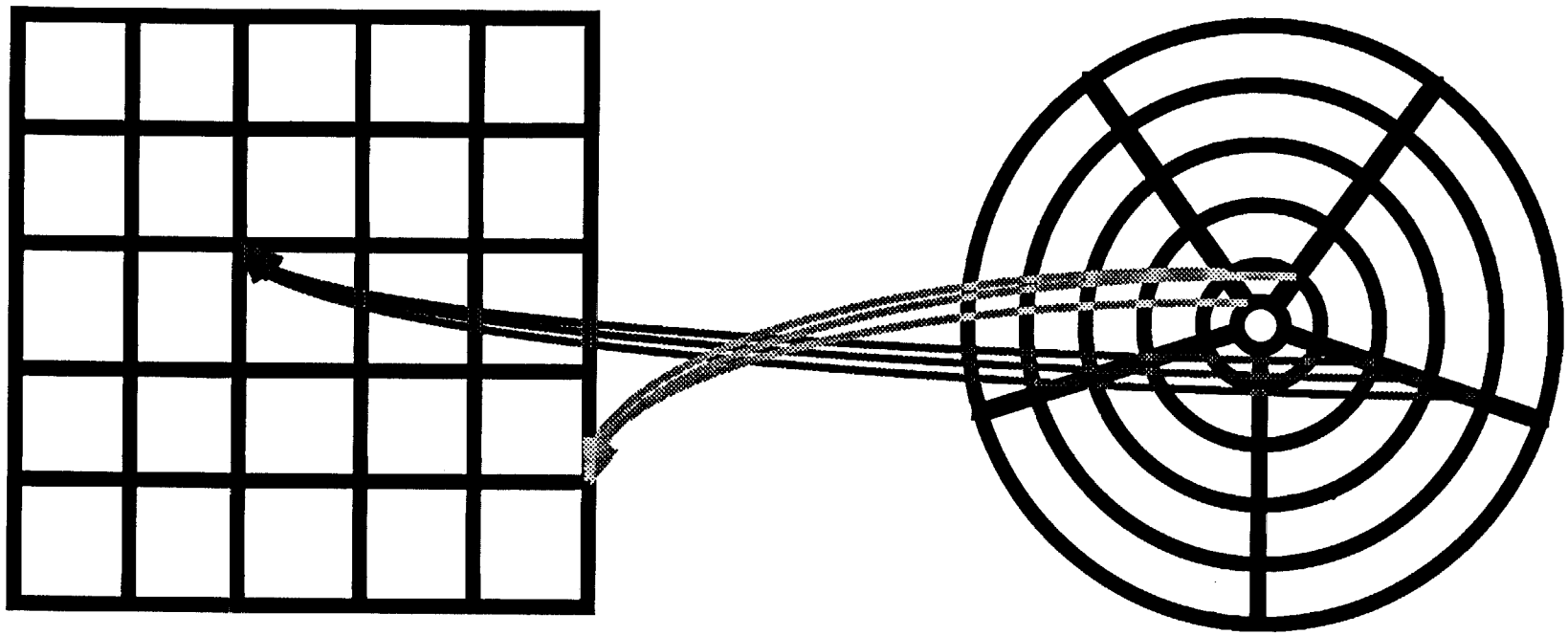


Figure 3. Weighted Average Gridding

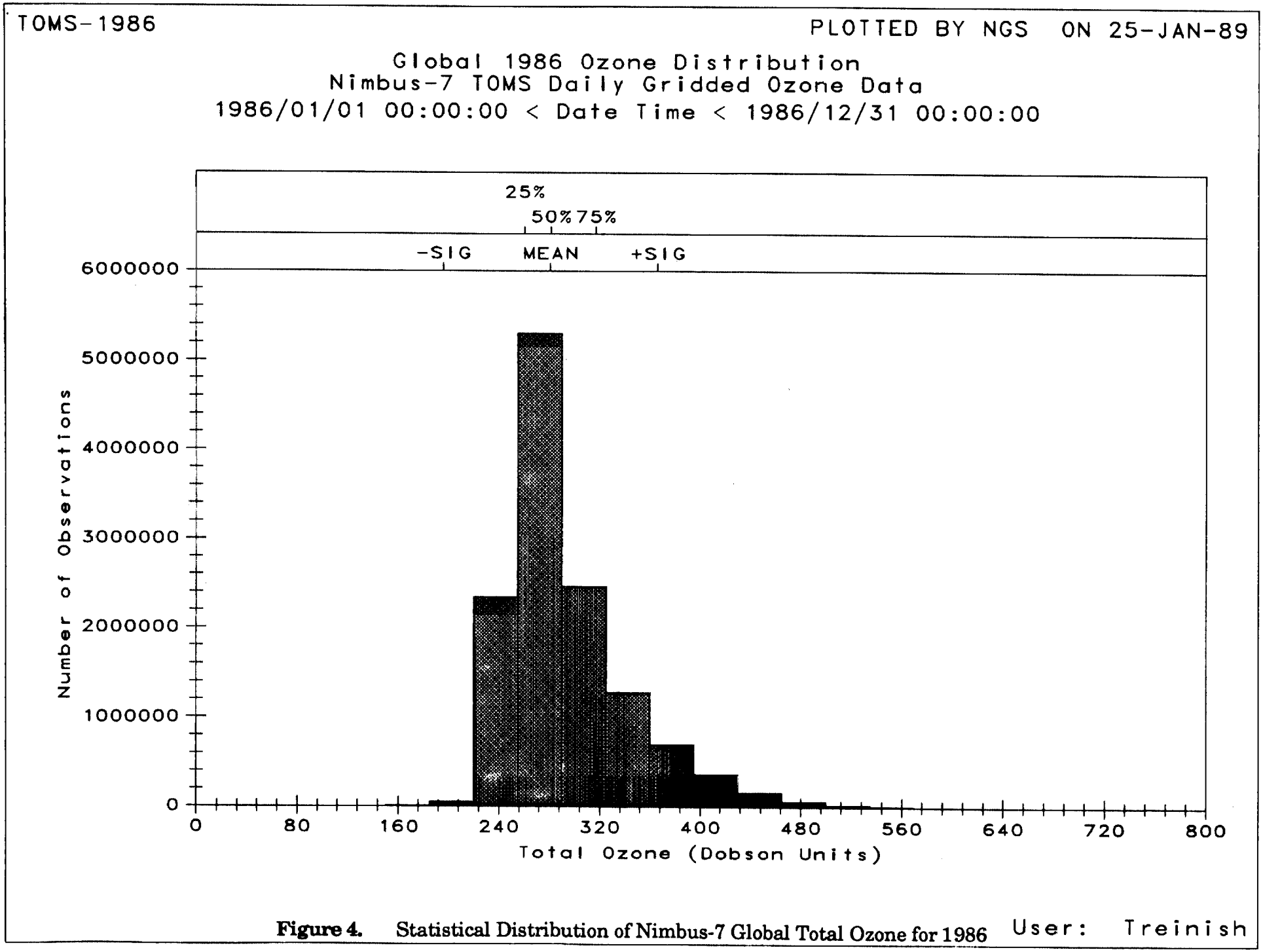


Figure 4. Statistical Distribution of Nimbus-7 Global Total Ozone for 1986

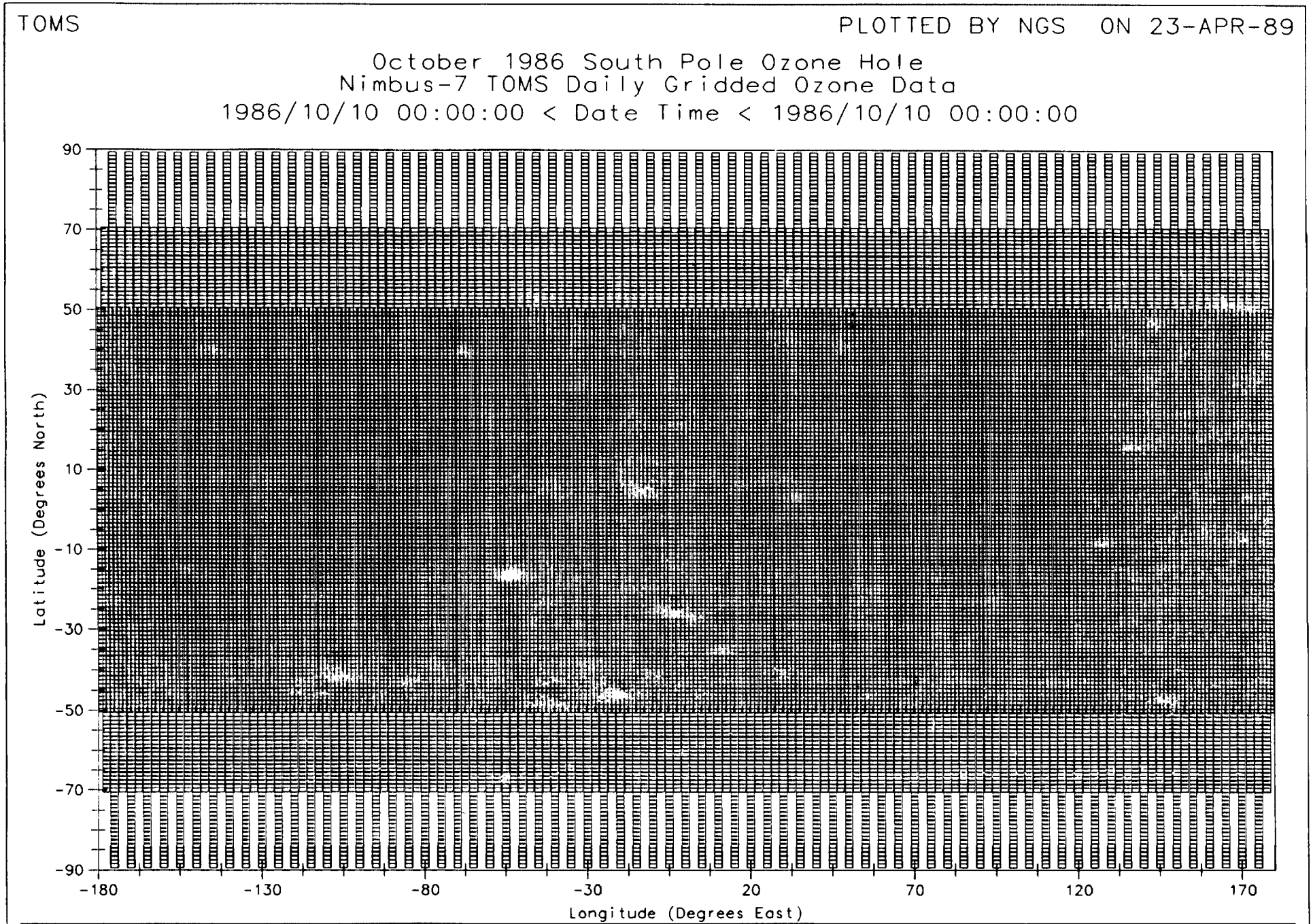


Figure 5. Spatial Distribution for Nimbus-7 TOMS Grids

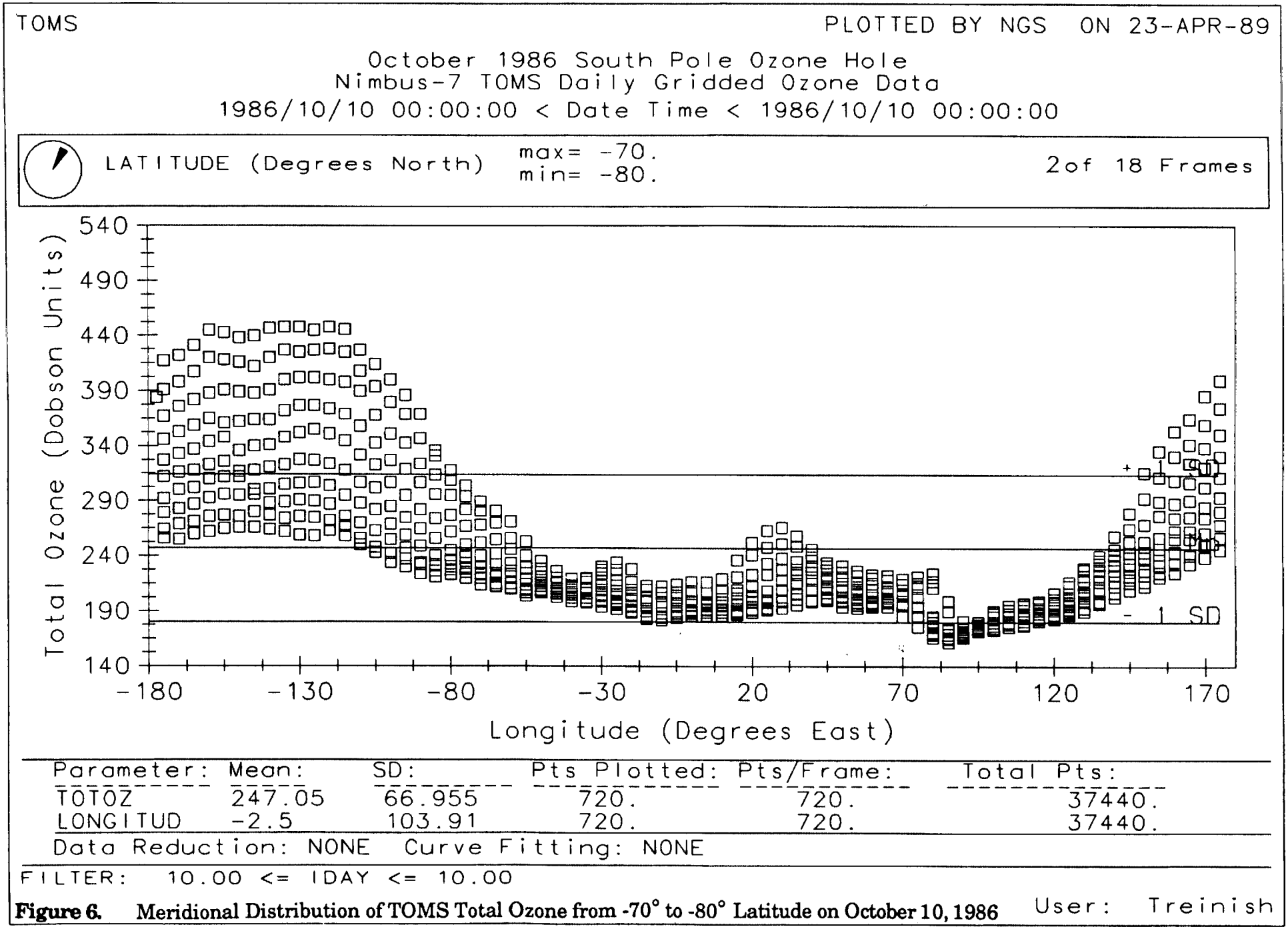


Figure 6. Meridional Distribution of TOMS Total Ozone from -70° to -80° Latitude on October 10, 1986

Figure 7. Total Global Ozone on October 10, 1986 on a Cartesian Map

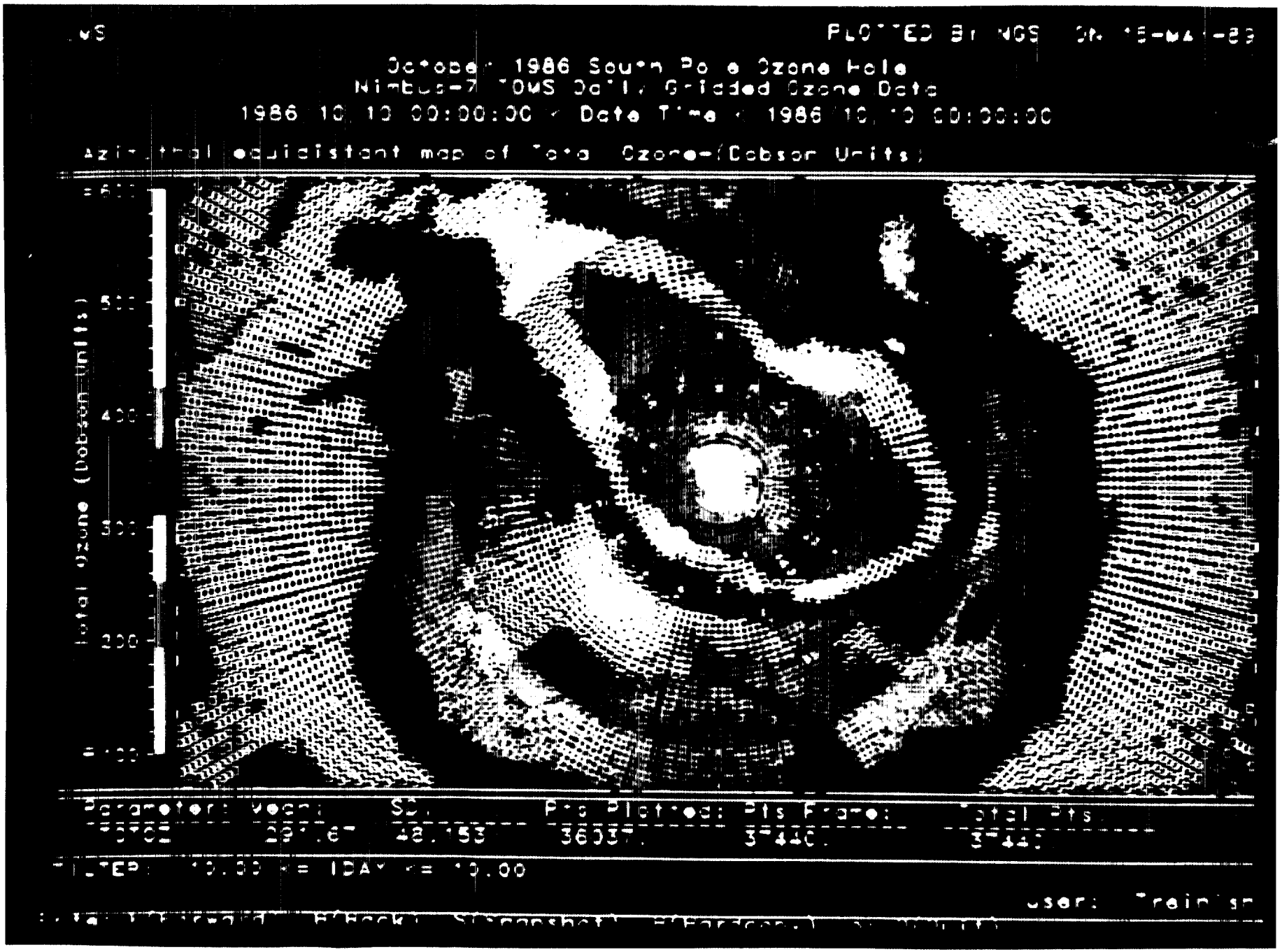


Figure 8. Spatial Distribution of Ozone over the Southern Hemisphere on October 10, 1986

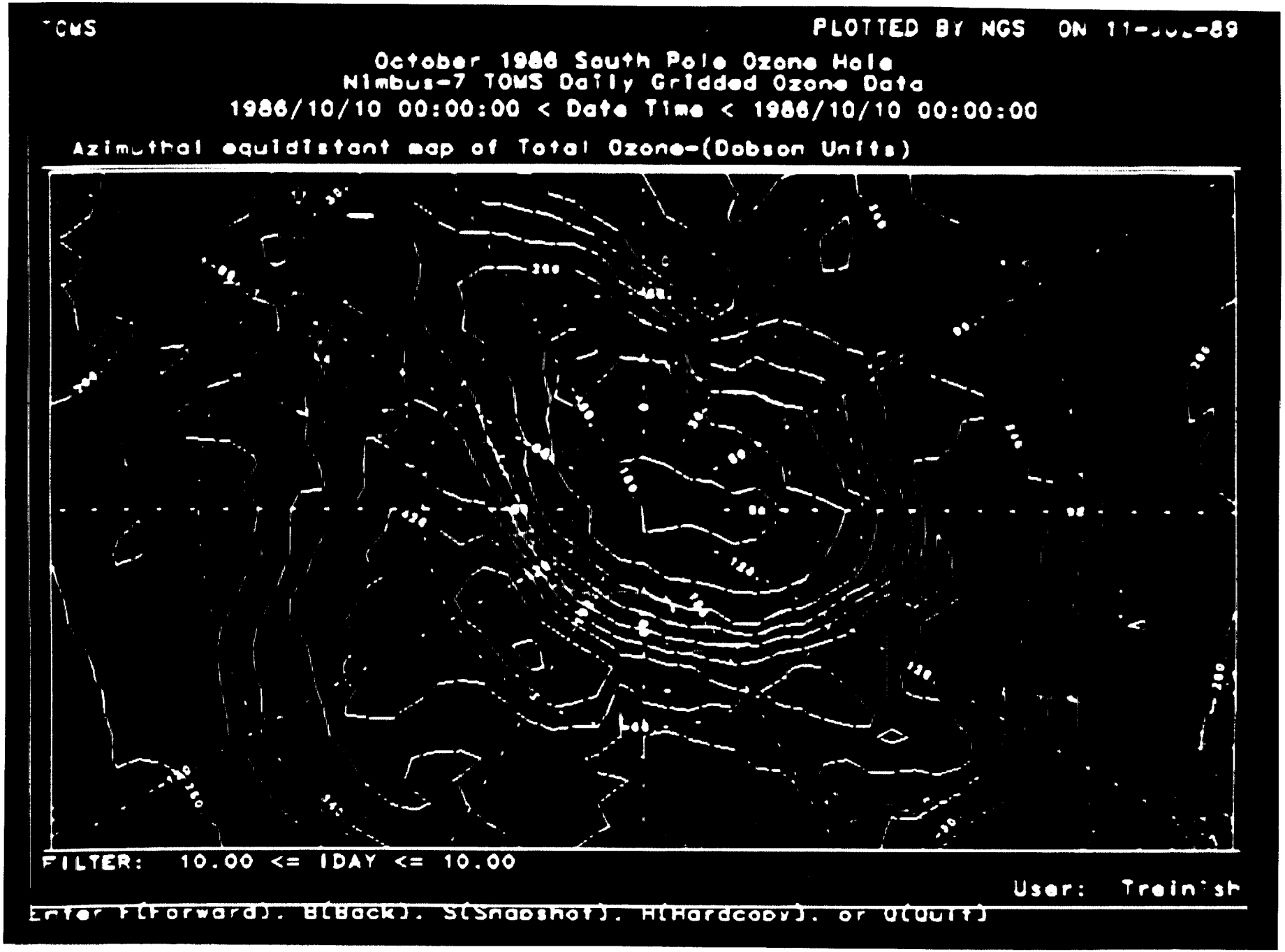


Figure 9. Simple Contour Map of Ozone over the Southern Hemisphere on October 10, 1986

Figure 10. Simple Mapped Pseudo-Color Image of Ozone over the Southern Hemisphere on October 10, 1986

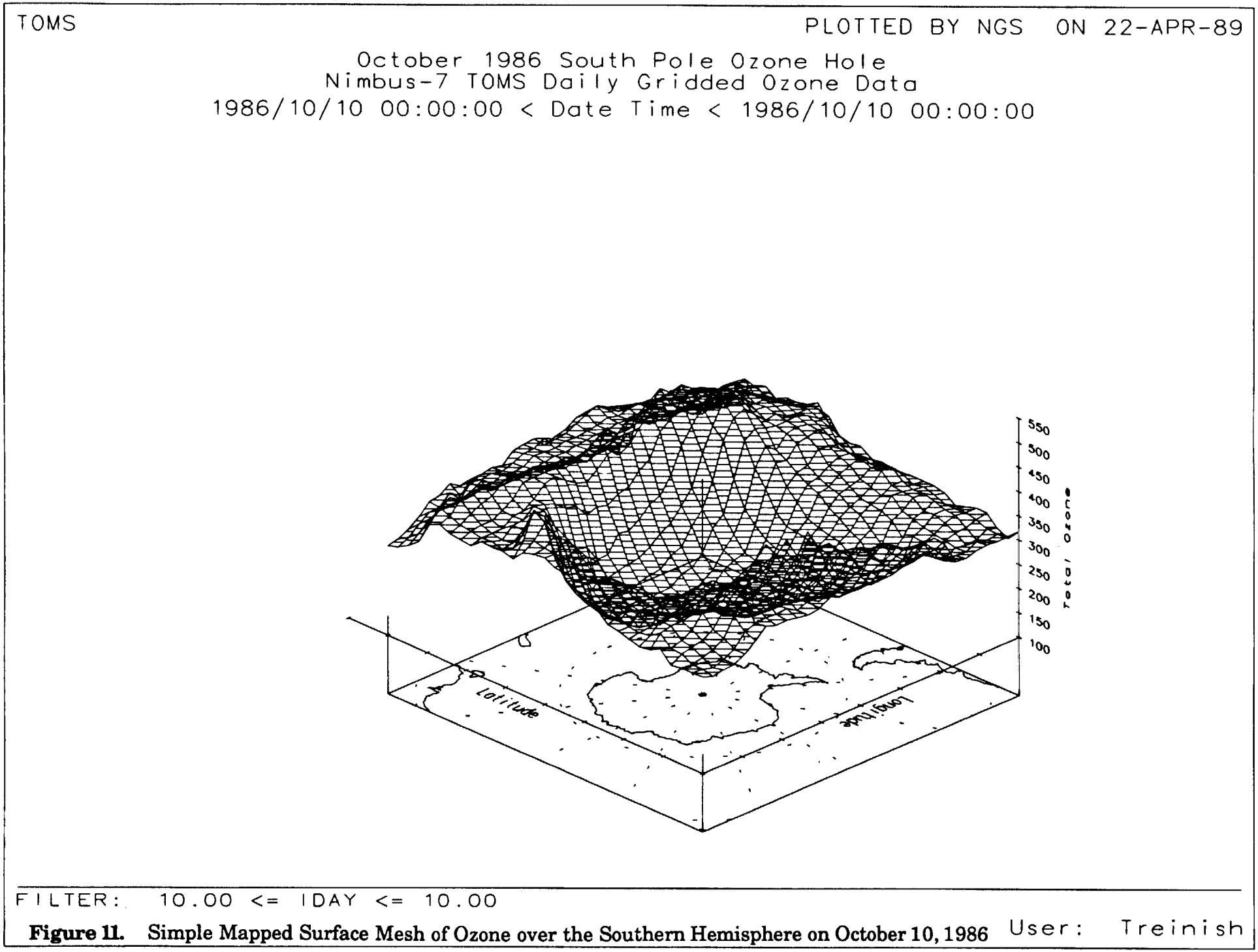


Figure 11. Simple Mapped Surface Mesh of Ozone over the Southern Hemisphere on October 10, 1986

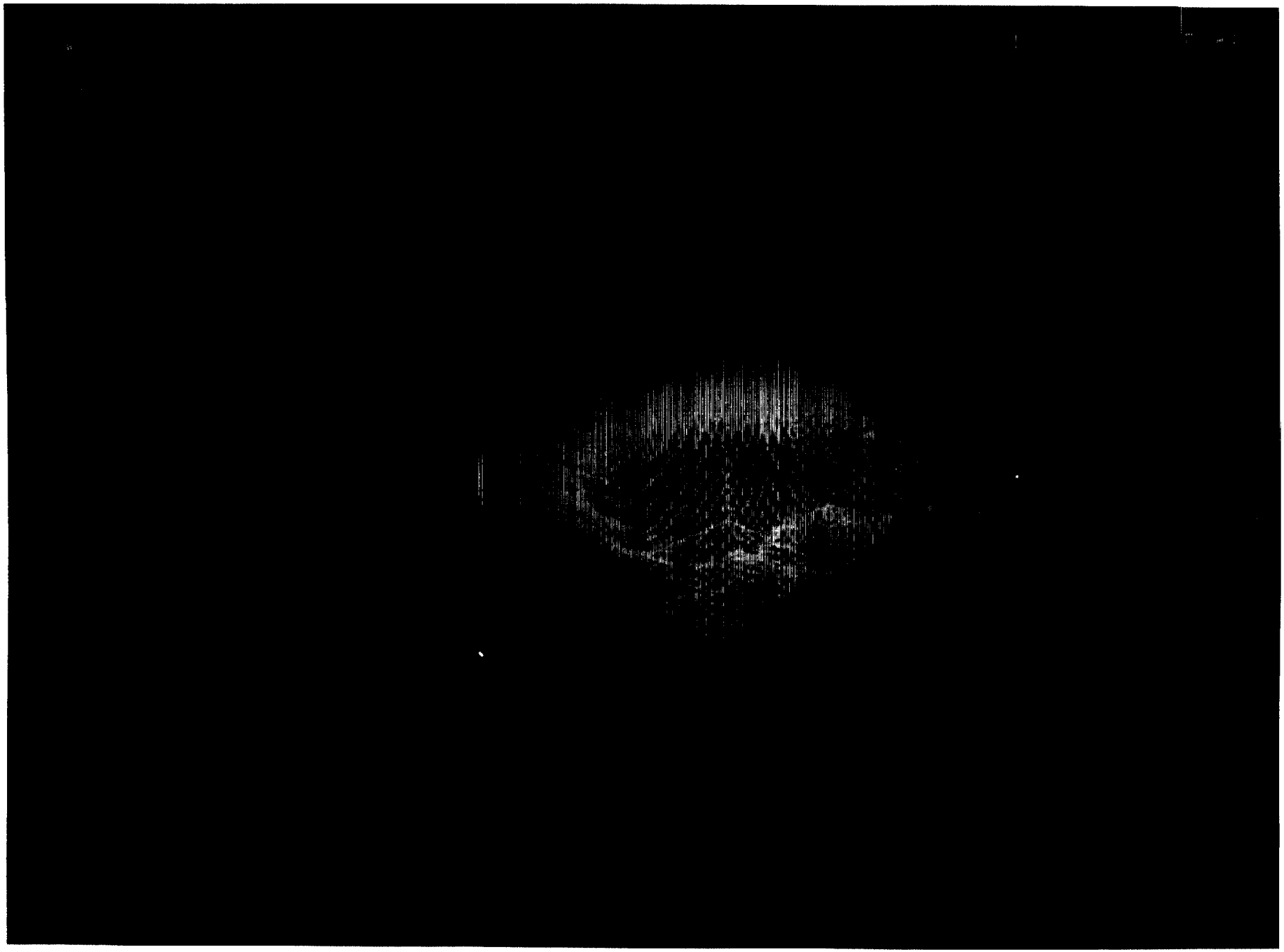


Figure 12. Mapped Shaded Surface Mesh of Ozone over the Southern Hemisphere on October 10, 1986

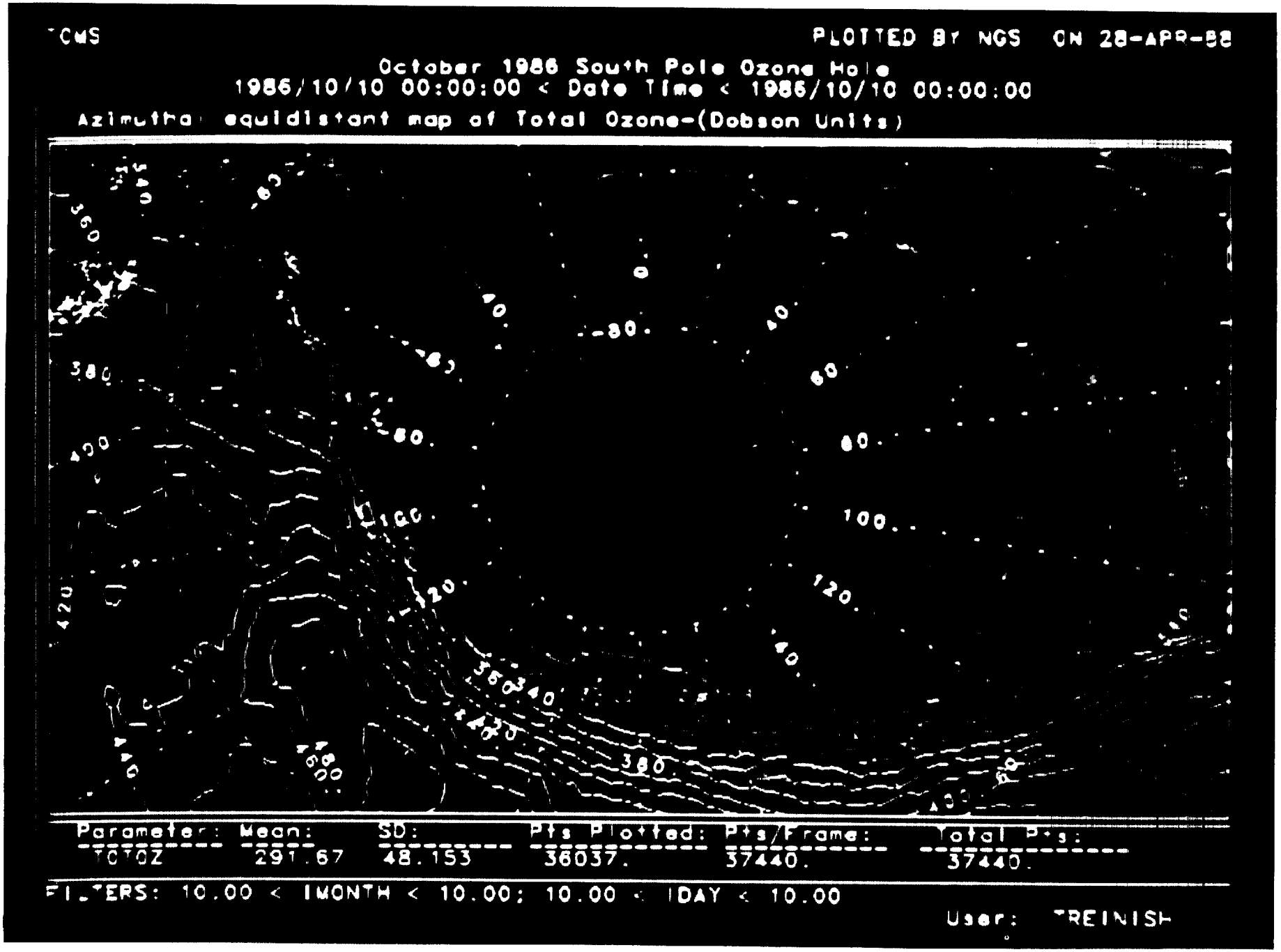


Figure 13. Optimized Contour Map of Total Ozone over Antarctica on October 10, 1986

Figure 14. Optimized Mapped Pseudo-color Image of Ozone over the Southern Hemisphere on October 10, 1986

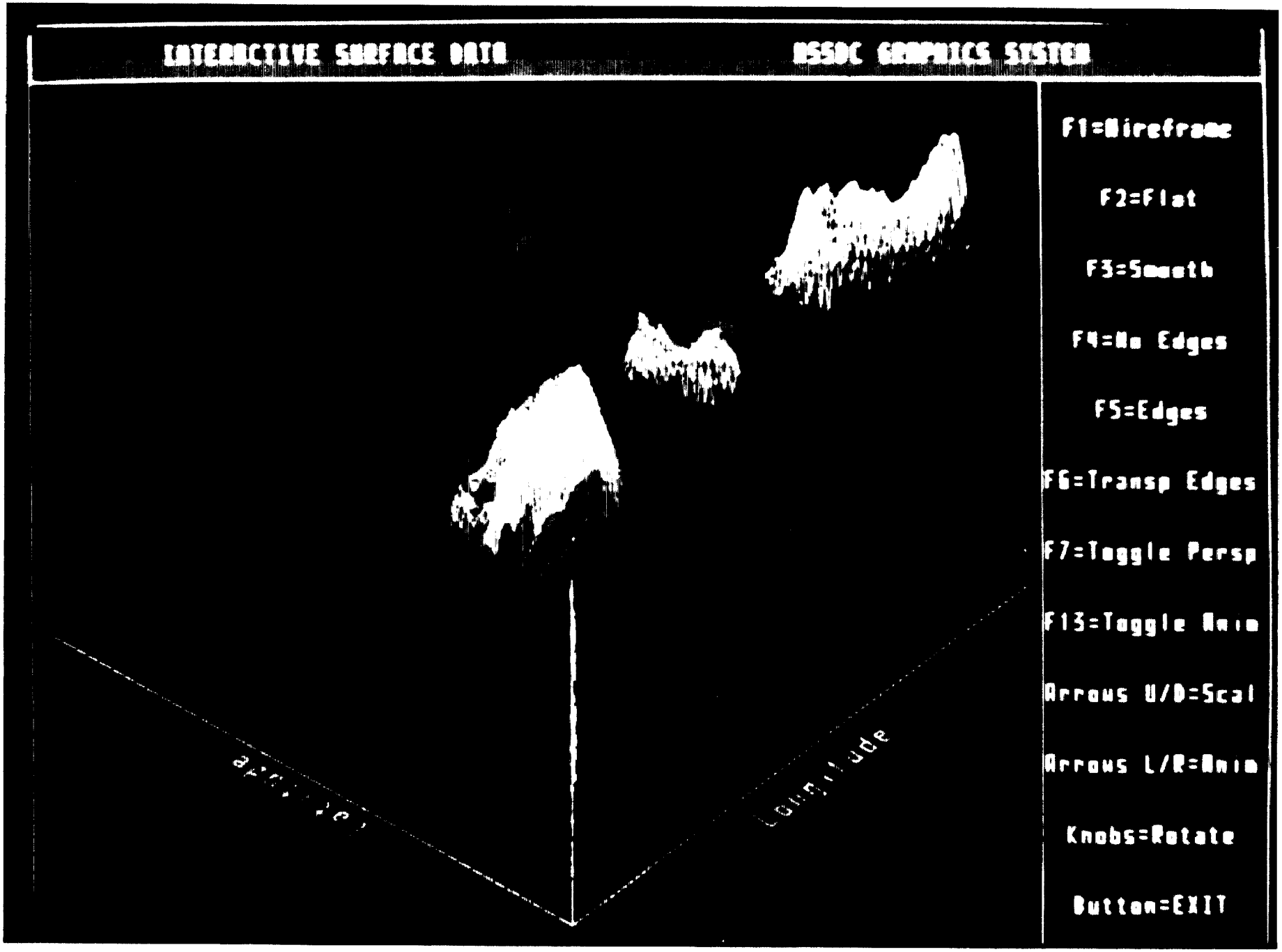


Figure 15. Cartesian Surface Mesh of Total Global Ozone on October 10, 1986

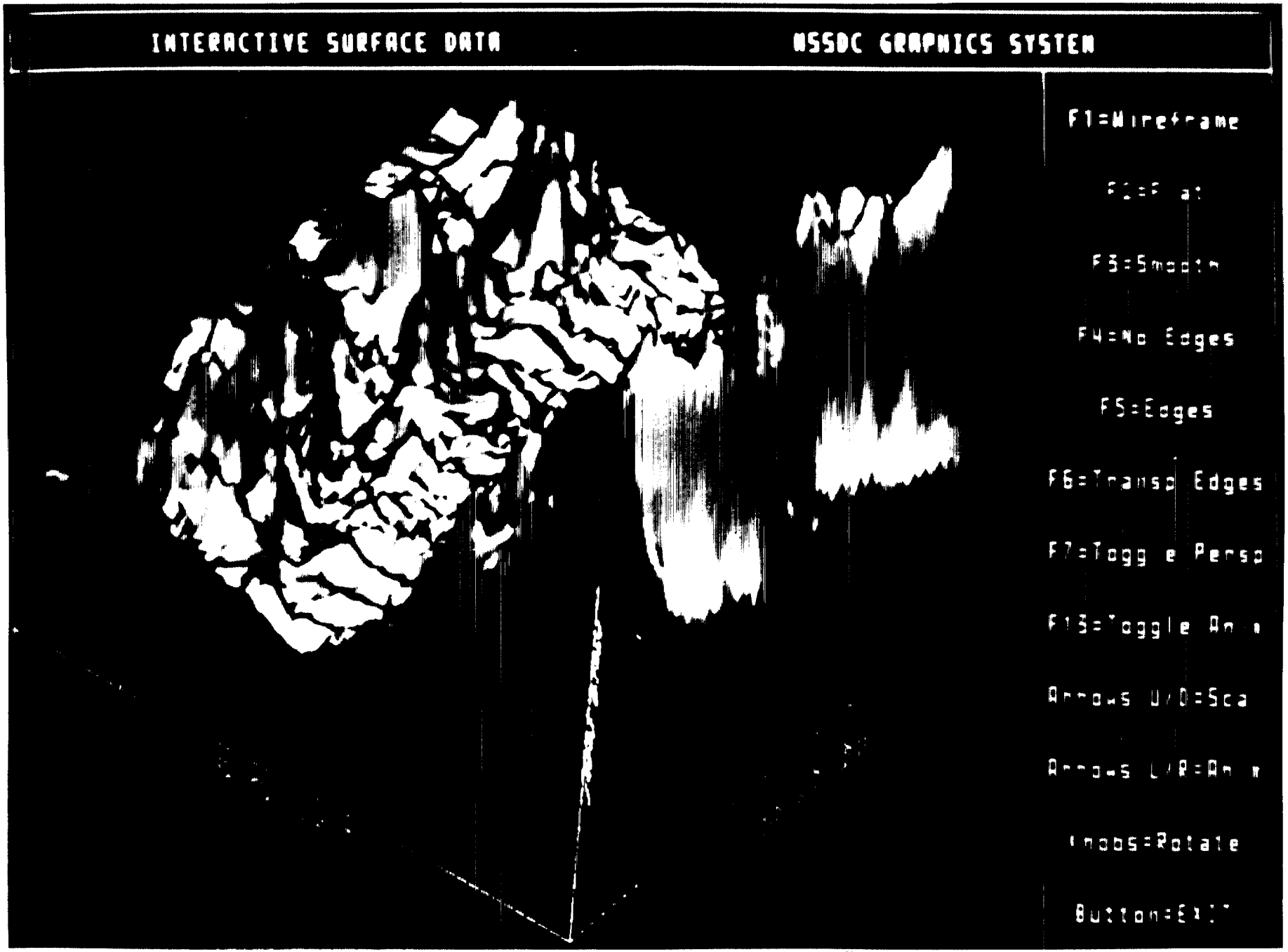


Figure 16. Gouraud-Shaded Cartesian Surface Mesh (137 x 137) of Total Global Ozone on October 10, 1986

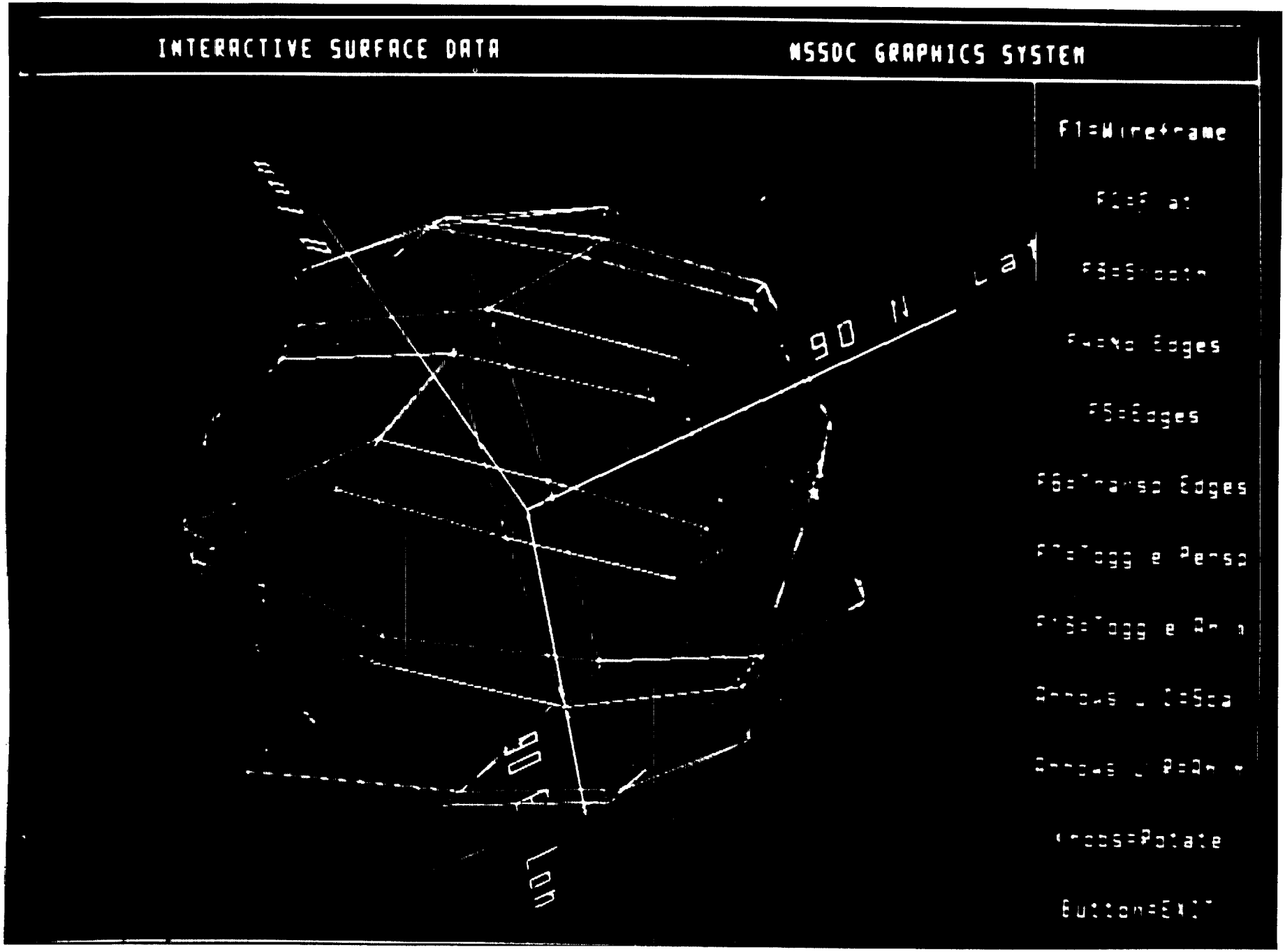


Figure 17. Geodesic Wire-Frame Model (80 triangles) of Total Global Ozone on October 10, 1986

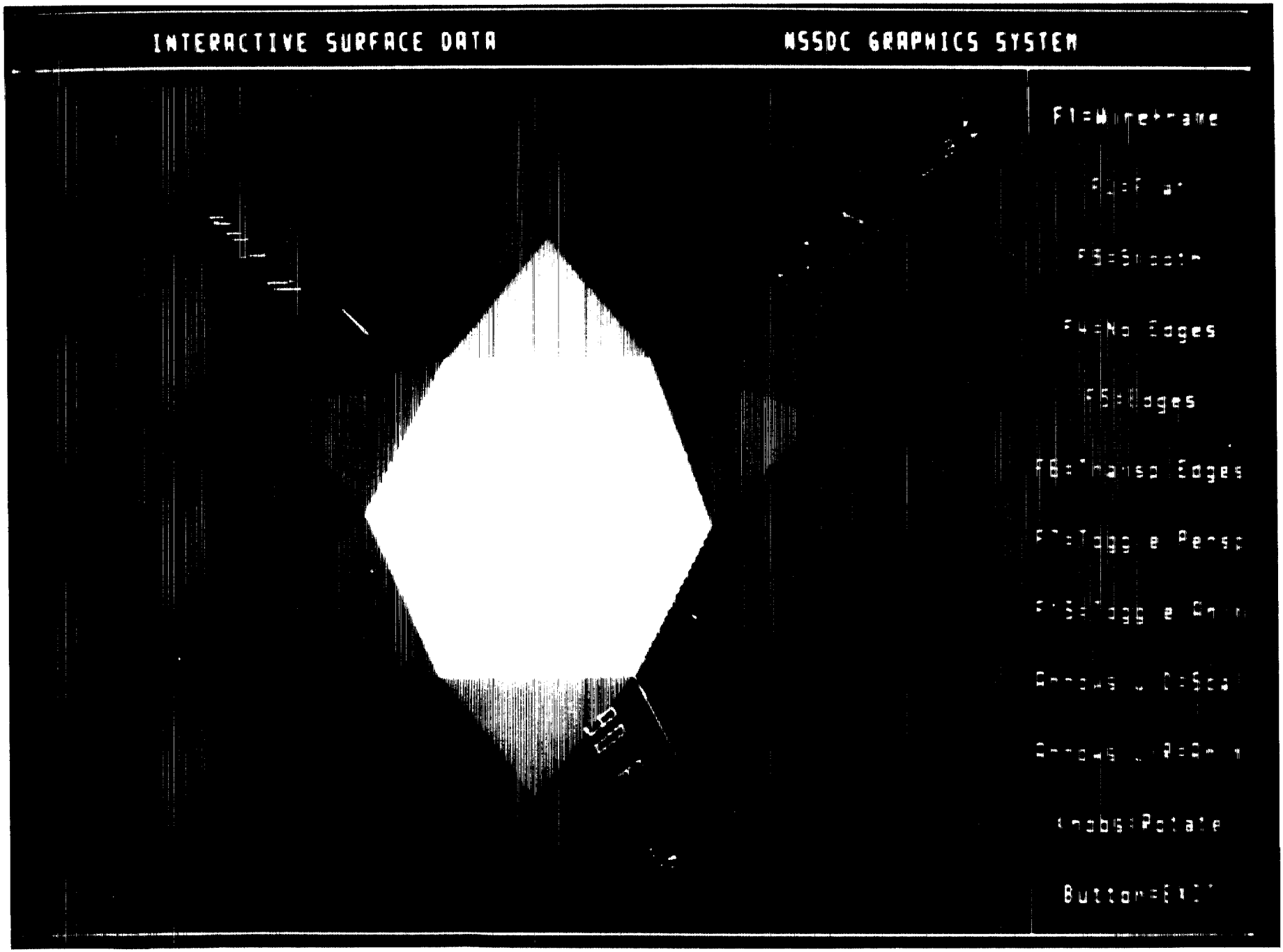


Figure 18. Geodesic Shaded Model (80 triangles) of Total Global Ozone on October 10, 1986

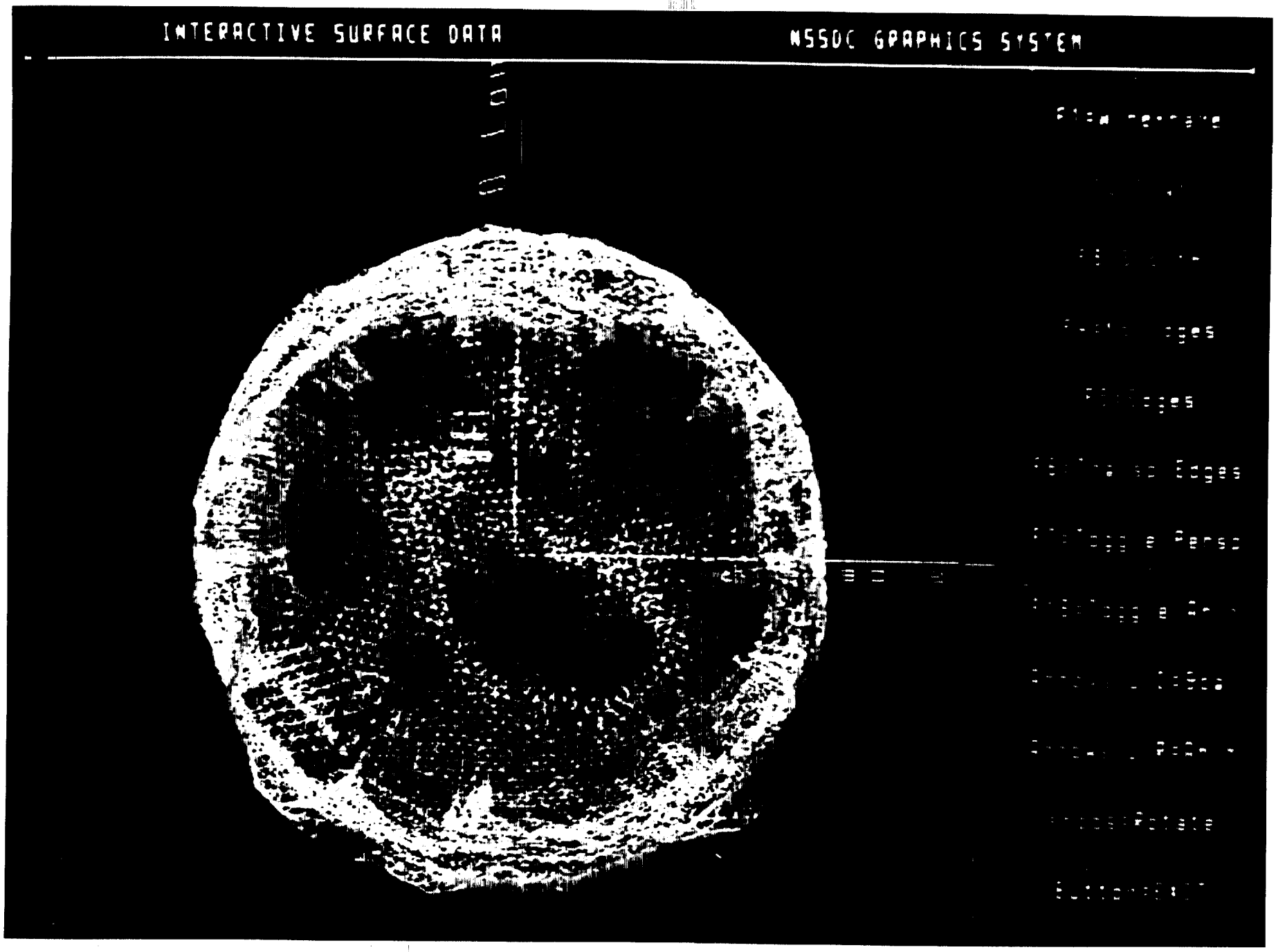


Figure 19. Geodesic Wire-Frame Model (20,480 triangles) of Total Global Ozone over the South Pole on October 10, 1986

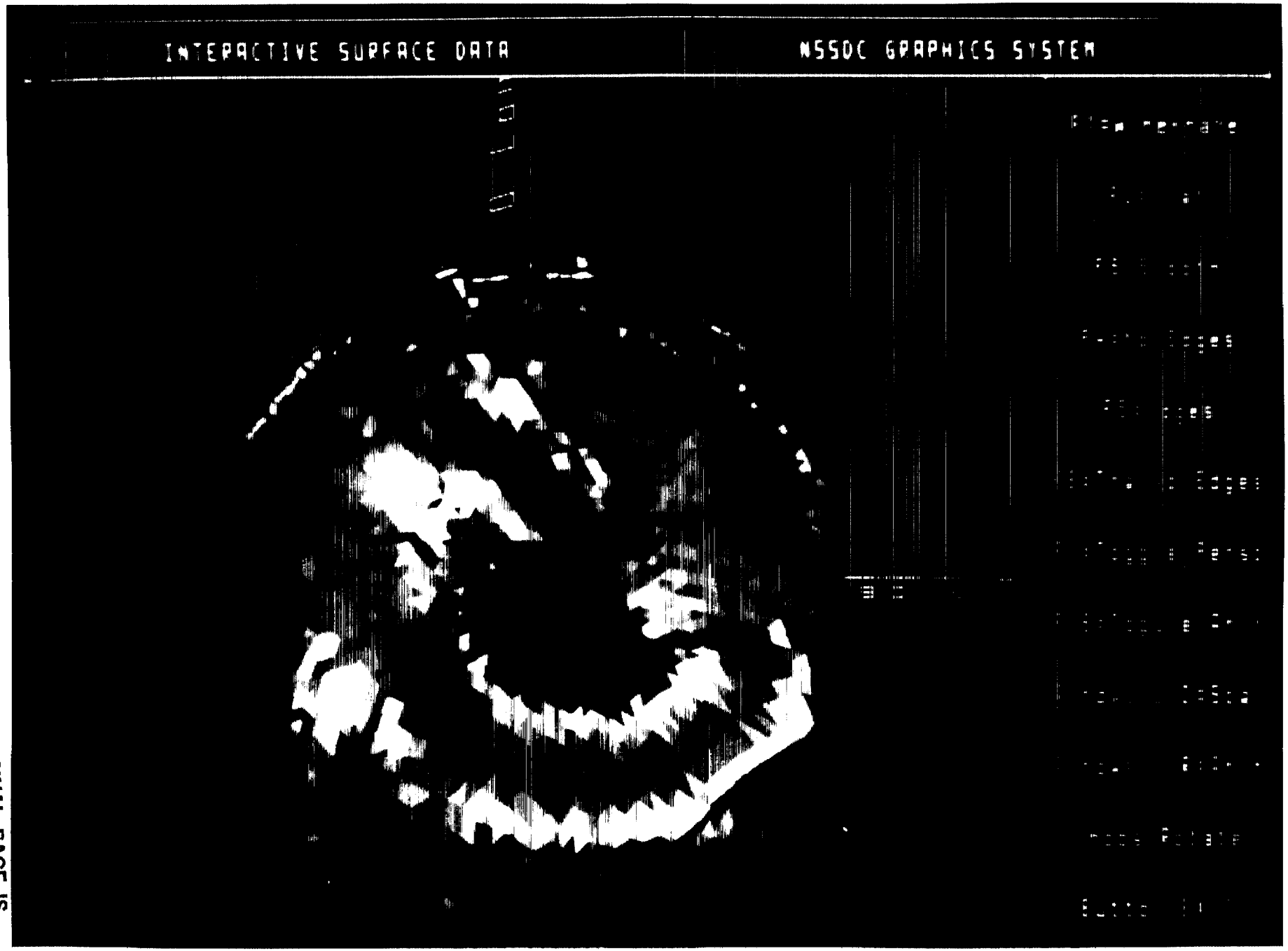


Figure 20. Geodesic Shaded Model (20,480 triangles) of Total Global Ozone over the South Pole on October 10, 1986

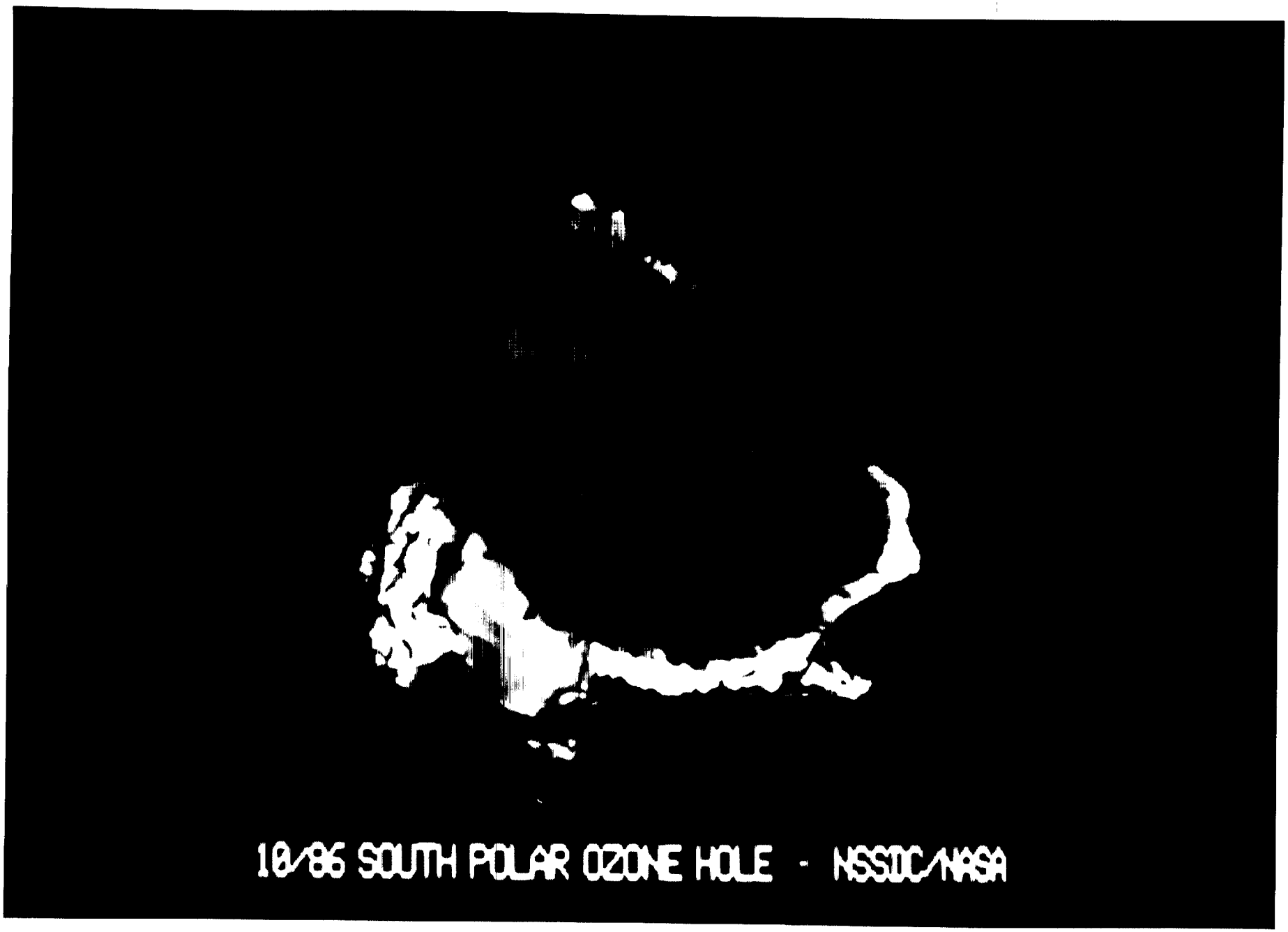


Figure 21. Ray-traced Rendering of Geodesic Model (81,920 triangles) of Total Global Ozone over the South Pole on October 10, 1986

Figure 22. Pseudo-color Image of the Zonal Distribution of Nimbus-7 TOMS Total Ozone in 1986

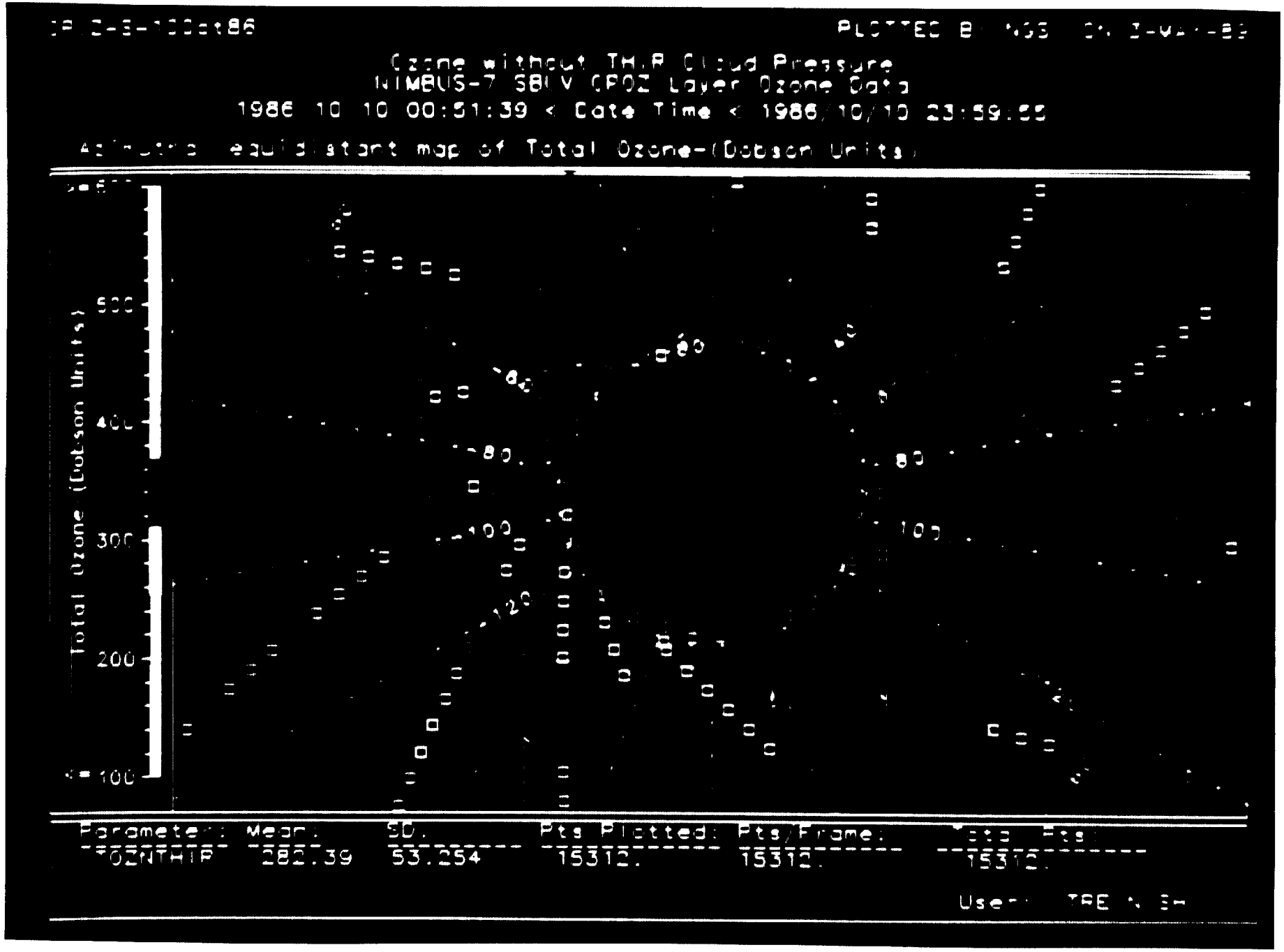
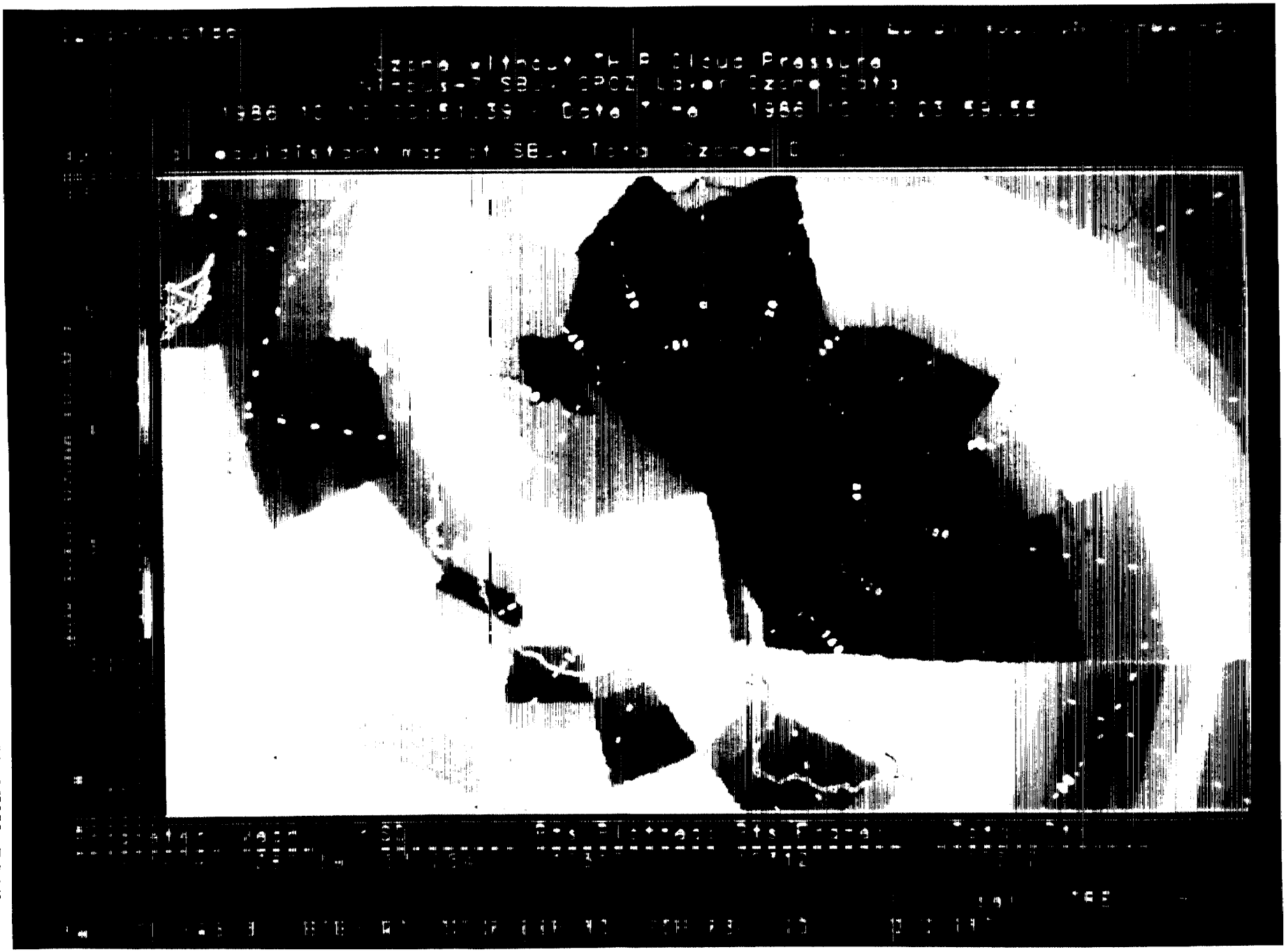


Figure 23. Spatial Distribution of Nimbus-7 SBUV Total Ozone over Antarctica on October 10, 1986



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 24. Map of SBUV Total Ozone over Antarctica on October 10, 1986

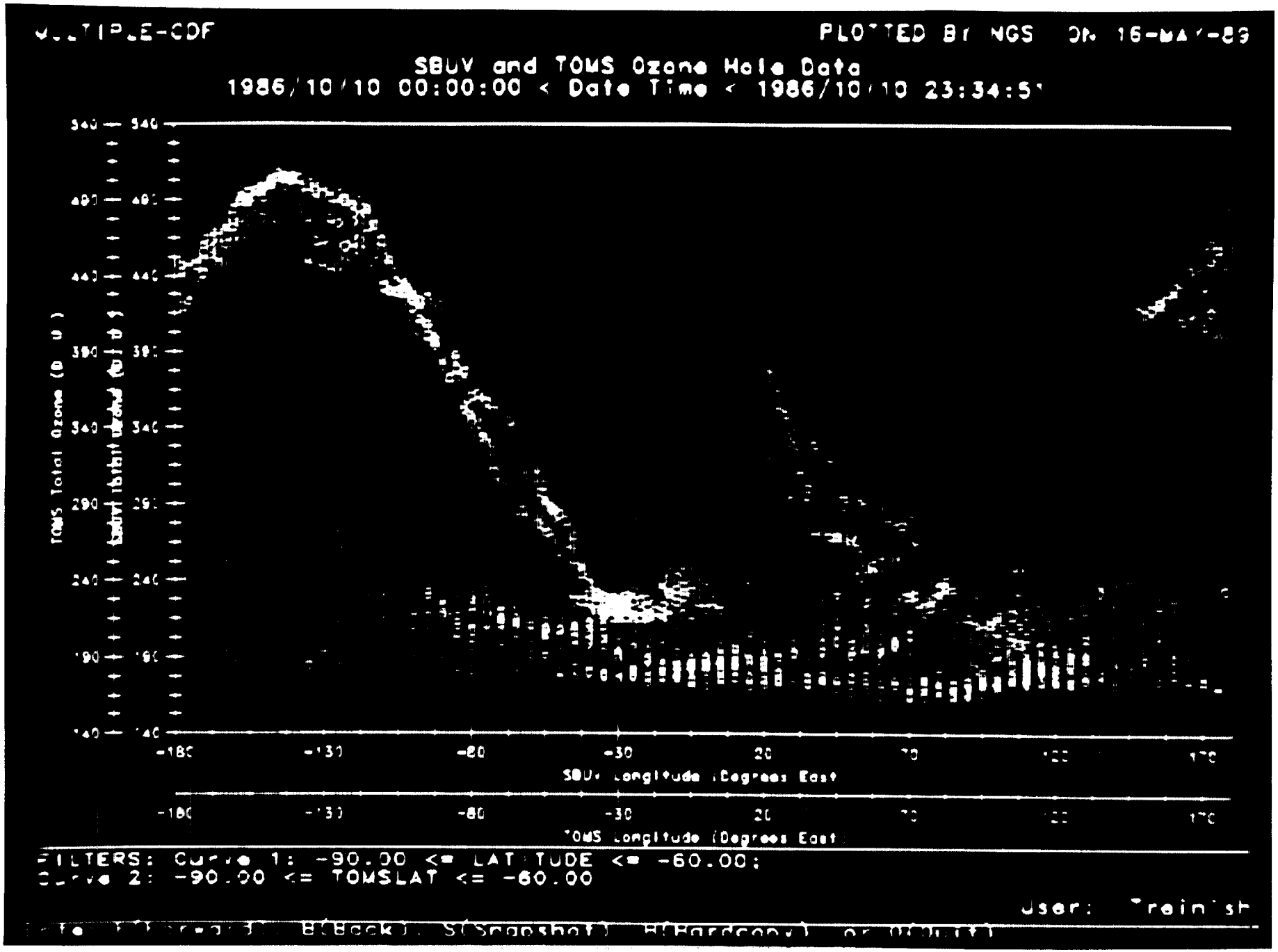


Figure 25. Meridional Distribution of TOMS and SBUV Total Ozone from -60° to -90° Latitude on October 10, 1986

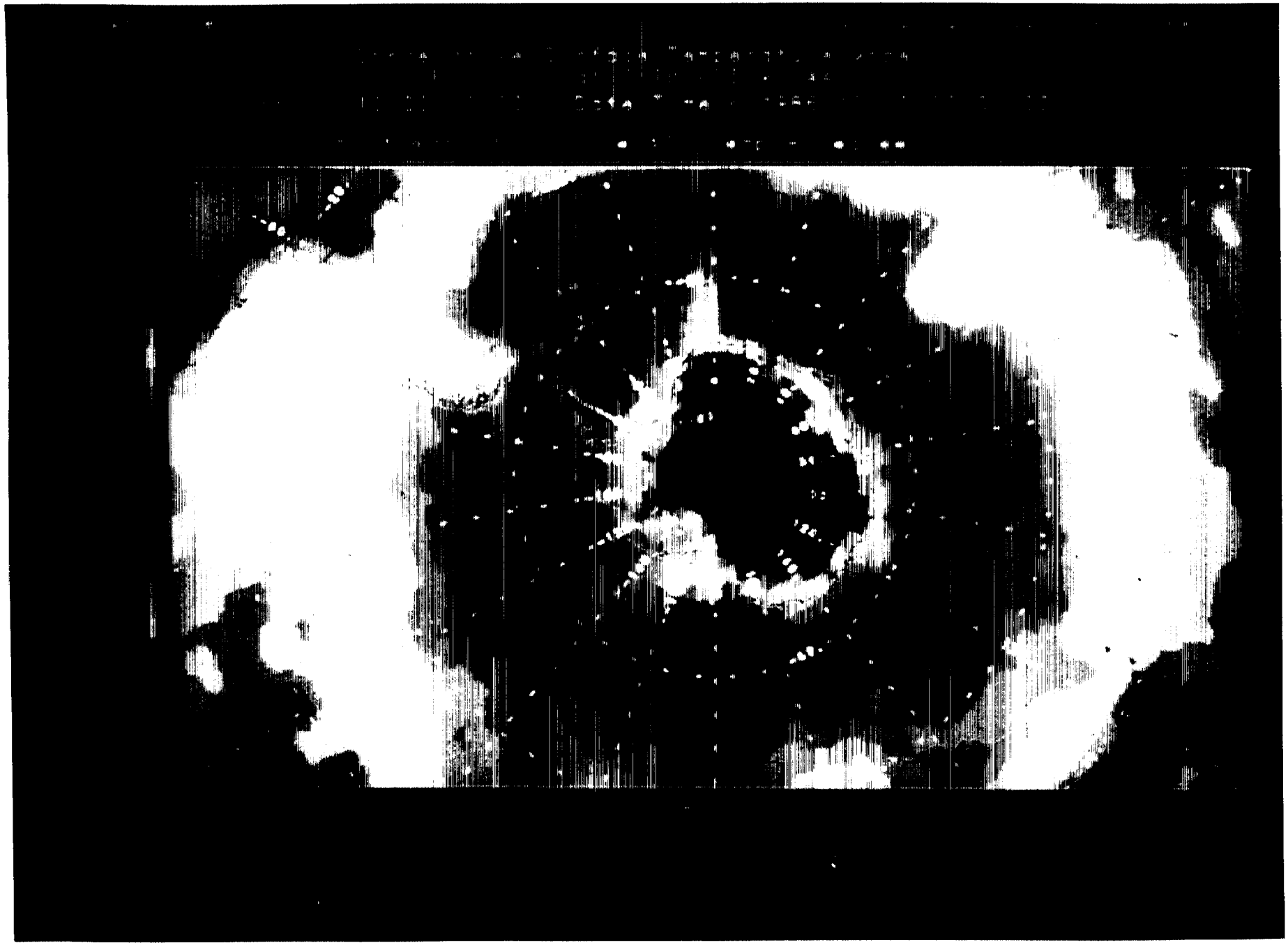


Figure 26. Map of FNOC Model Surface Temperature on October 10, 1986 over the Southern Hemisphere