NASA-CR-194489

GRANT
IN 90-CR
186481

105 P

Visualization Techniques to Aid in the Analysis of Multi-Spectral Astrophysical Data Sets

Final Report on grant NAGW-1902
Prepared for the National Aeronautics and Space Administration
HQ, Astrophysics (Code SZ)

September 1993

Principal Investigator: E.W. Brugel, Ph.D.
Investigators: G.O. Domik (Lead-Investigator), Ph.D., and T.R. Ayres, Ph.D.

Report prepared by

Gitta Domik[1]

Work performed at the University of Colorado at Boulder
Center for Astrophysics and Space Astronomy &
Department of Computer Science

[1] Current address: University of Paderborn, Department of Mathematics and Computer Science, FB 17, D-33095 Paderborn, Germany
Tel. +49/5251/60-2605 Fax: 60-3836 email:domik@uni-paderborn.de

Please reference this document as:

Domik, G.O., 1993, *Visualization Techniques to Aid in the Analysis of Multi-Spectral Astrophysical Data Sets*, Final report to NASA, Astrophysics Division, on grant NAGW-1902. University of Colorado, Boulder, CO. 80309-0389.

Acknowledgements

The investigators wish to thank the Astrophysics Division of the National Aeronautics and Space Administration, specifically Dr. Günter Riegler, to give them the opportunity to investigate into the capabilities and challenges of Scientific Visualization in regards to astrophysical data.

The support of the Center for Astrophysics and Space Astronomy (CASA) in providing a stimulating scientific environment, hardware and software resources, and general administrative assistance is much appreciated. Many thanks go to Dr. Clayton Lewis, Department of Computer Science, and Dr. Theodore Snow, director of CASA, for consultancy and support during the project.

During 1989 and 1993 many contributions were made by members of the Center for Astrophysics and Space Astronomy as well as by members of the Department of Computer Science. We gratefully acknowledge the work of the following individuals:

S. Alam, G. Allison, R. Chaiken, W. Fortson, C. Joslyn, S. Merkle, K. Mickus-Miceli, P. Pinkney, W. Radinsky, R. Tarrall, A. Veale, J. Wilson.

Table of Contents

Executive Summary

The goal of this project was to support the scientific analysis of multi-spectral astrophysical data by means of scientific visualization. Scientific visualization offers its greatest value if it is not used as a method separate or alternative to other data analysis methods but rather in addition to these methods. Together with quantitative analysis of data, such as offered by statistical analysis, image or signal processing, visualization attempts to explore *all* information inherent in astrophysical data in the most effective way.

Data visualization is one aspect of data analysis. Our taxonomy as developed in Section 2 includes identification and access to existing information, preprocessing and quantitative analysis of data, visual representation and the user interface as major components to the software environment of astrophysical data analysis. In pursuing our goal to provide methods and tools for scientific visualization of multi-spectral astrophysical data, we therefore looked at scientific data analysis as one whole process, adding visualization tools to an already existing environment and integrating the various components that define a scientific data analysis environment. As long as the software development process of each component is separate from all other components, users of data analysis software are constantly interrupted in their scientific work in order to convert from one data format to another, or to move from one storage medium to another, or to switch from one user interface to another.

We also took an in-depth look at scientific visualization and its underlying concepts, current visualization systems, their contributions and their shortcomings. The role of data visualization is to stimulate mental processes different from quantitative data analysis, such as the perception of spatial relationships or the discovery of patterns or anomalies while browsing through large data sets. Visualization often leads to an intuitive understanding of the meaning of data values and their relationships by sacrificing accuracy in interpreting the data values. In order to be accurate in the interpretation, data values need to be measured, computed on, and compared to theoretical or empirical models (quantitative analysis). If visualization software hampers quantitative analysis (which happens with some commercial visualization products), its use is greatly diminished for astrophysical data analysis.

The software system STAR (Scientific Toolkit for Astrophysical Research) was developed as a prototype during the course of the project to better understand the pragmatic concerns raised in the project. STAR led to a better understanding on the importance of collaboration between astrophysicists and computer scientists.

Twenty-one examples of the use of visualization for astrophysical data are included with this report. Sixteen publications related to efforts performed during or initiated through work on this project are listed at the end of this report.

1. Introduction

1.1 Definitions

“Visualization of Astrophysical Data” describes the application of graphical methods to enhance interpretation and meaning of data measured or computed to gain insight into scientific questions to be answered by astrophysicists. The goals of visualization go beyond the mere use of tools offered by computer graphics systems: visualization is directed towards the use of appropriate graphical methods to enhance current understanding of scientific data.

“Multi-spectral data” (or multi-wavelength, multi-variate data) describe celestial objects through a range of observations over the electromagnetic spectrum. This approach is often essential in the development of a complete physical model of an astronomical source. For example, to understand the energy budget of a cool star having a chromosphere and corona, it is necessary to measure broad-band fluxes in the X-ray and radio portions of the spectrum, as well as to acquire moderate-resolution emission line profiles in the ultraviolet and visible regions. To study the relationship between interstellar gas abundances and the kinematics of discrete clouds, equivalent widths from ultraviolet spectra strongly complement super-high-resolution optical spectroscopy. To properly attack the question of the central powerhouse of quasars and active galactic nuclei, it is imperative to record the energy distributions - and their variability - over virtually the entire electromagnetic spectrum.

The list of astrophysical problems best treated with multispectral analysis is nearly as long as the list of all astronomical research objectives today. The concept of multispectral astronomy is hardly new, but the tools with which to implement it have been lacking. The observational side of multispectral astronomy is being addressed with the new generation of space observatories of the 1990’s and beyond; the analysis side -- software tools and environments -- is less developed. Our project sought to redress this lack by inspecting currently available software for the analysis and visualization of multispectral data and designing new software where shortcomings exist.

STAR (Scientific Toolkit for Astrophysical Research) is the software developed in the course of the project. STAR was developed as a prototype to prove the feasibility of user interface, software engineering and visualization techniques suggested in this report.

1.2 Goals of the project

The goal of this project was to support the scientific analysis of multi-spectral astrophysical data by means of scientific visualization. Scientific visualization offers its greatest value if it is not used as a method separate or alternative to other data analysis methods but rather in addition to these methods. Together with quantitative analysis of data, such as offered by statistical analysis, image or signal processing, visualization attempts to explore *all* information inherent in astrophysical data in the most effective way. Visualization is a vehicle of thinking (McKim, 1980), capable to explore spatial relationships between data items, making use of intuitive and holistic approaches to reasoning and the effortless identification of patterns and anomalies in large data sets. A scientist is in need of a multitude of methods and tools to explore *all* aspects of scientific data. It is important that all tools are integrated with each other and with the already existing environment of scientific data analysis.

In pursuing our goal to provide methods and tools for scientific visualization of multi-spectral astrophysical data, we therefore looked at scientific data analysis as one whole process, adding visualization tools to an already existing environment and integrating the various components that define a scientific data analysis environment. From this view the following work has emerged. With our work we hope to have added to a better understanding of the capabilities and challenges of Scientific Visualization in regards to astrophysical data analysis.

In the next chapter we first describe a taxonomy for data analysis. We then describe the processes involved with each analysis component in more detail. Chapter 3 lists examples of work performed under the grant. Chapter 4 recounts the history of work progress between 1989 and 1993. The final two chapters include recommendations to the sponsors of this project and a list of publications which derived from work on this project.

2. Astrophysical Data Analysis

2.1 Data analysis taxonomy

A data analysis taxonomy was developed while writing the proposal for this project (Ayres, Brugel, Domik and 1989) and is shown in Figure 1:

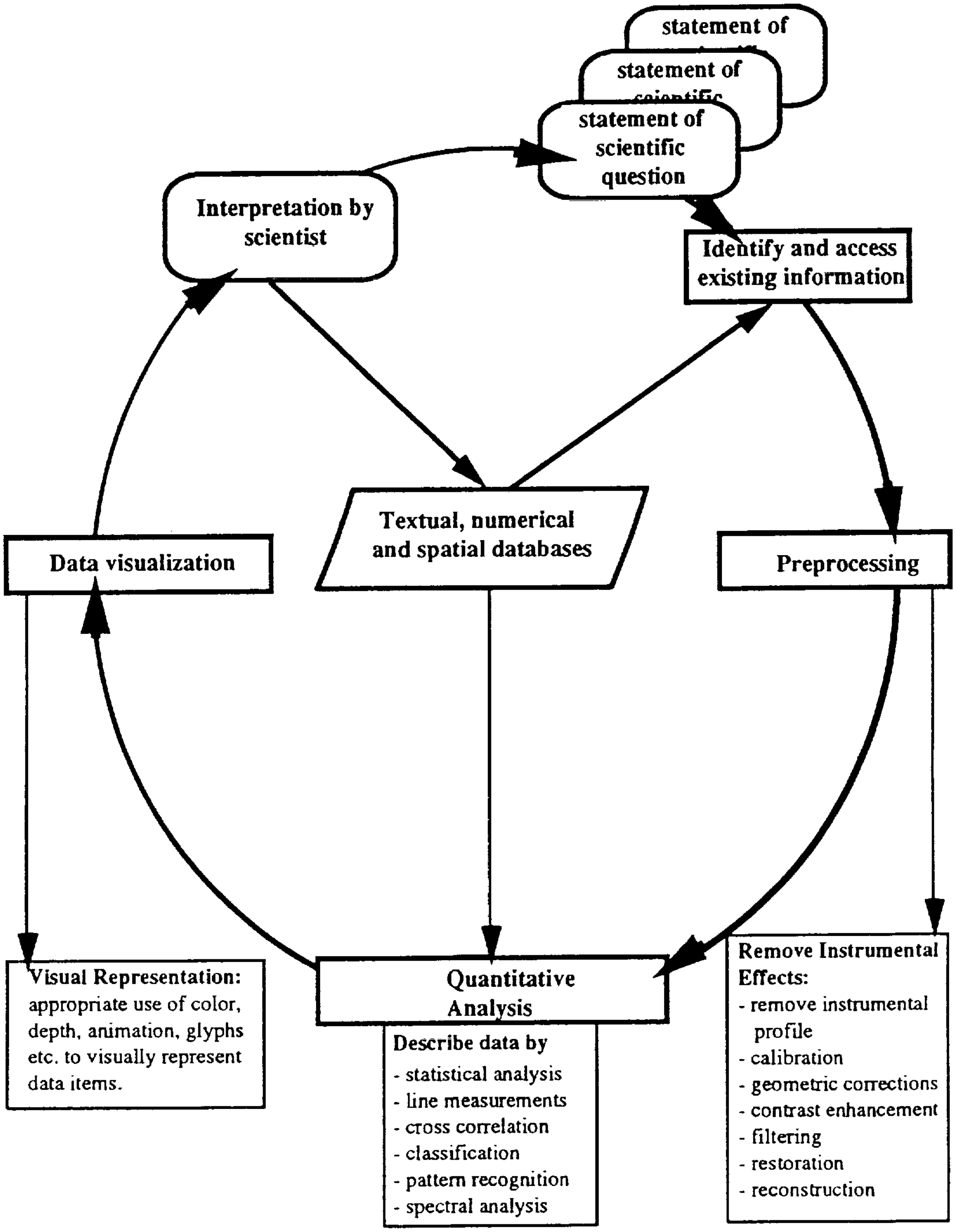

statement of
statement of
statement of scientific question
Interpretation by scientist
Identify and access existing information
Textual, numerical and spatial databases
Data visualization
Preprocessing
Visual Representation: appropriate use of color, depth, animation, glyphs etc. to visually represent data items.
Quantitative Analysis
Describe data by
- statistical analysis
- line measurements
- cross correlation
- classification
- pattern recognition
- spectral analysis
Remove Instrumental Effects:
- remove instrumental profile
- calibration
- geometric corrections
- contrast enhancement
- filtering
- restoration
- reconstruction

Figure 1: Taxonomy of Data Analysis.

This data analysis taxonomy provides a reference model to the environment of an astrophysicist utilizing multi-spectral data:

Besides the four domains of data analysis (identify and access existing information; preprocessing; quantitative analysis; visual representation) as defined above, we also include the user interface as a major component to the software environment of astrophysical data analysis. These five domains are explained in more detail below.

While a deeper understanding of the role of data visualization in the analysis process of scientific data evolved over the past years of project work, we are still using our original taxonomy with only minor modifications. We see this as a validation of our original model over the course of the project.

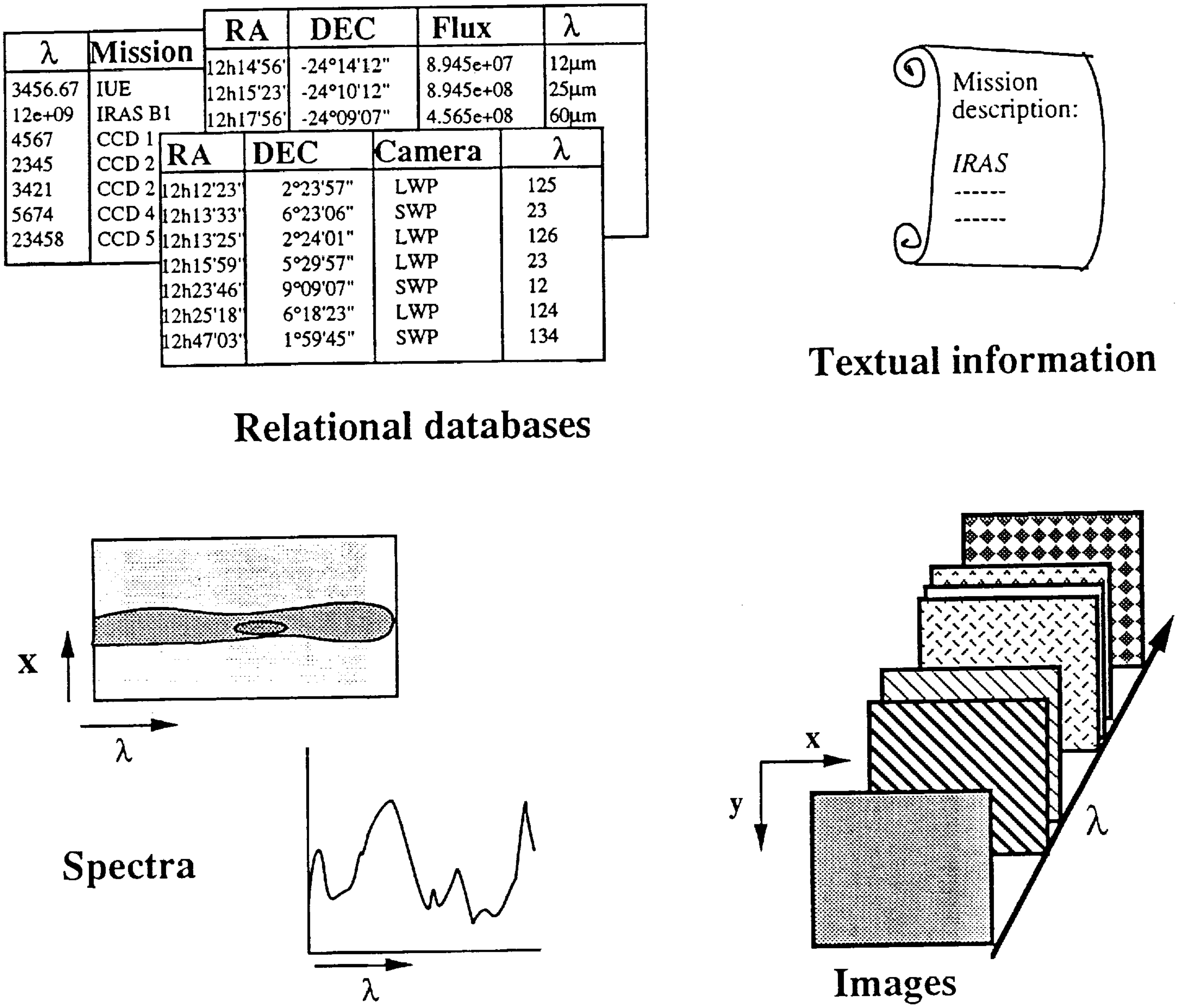
2.2 Identify and access existing information

The first task of the analysis system is to retrieve existing information and data pertinent to answering scientific questions of interest. Such information will come in various forms and structures such as images, spectra, tables or text (Figure 2). Some data will have spatial relevance, some will not. Access to different types of data bases (image, relational, textual; spatial and non-spatial) must be ensured so that a complete inventory of relevant information is presented to the scientists.

In order to improve the selection process for astrophysicists, information must be available at their fingertips. In STAR, efforts to improve the selection process were twofold: firstly, to provide access to existing data bases and secondly, to create new functionality for database selections.

2.2.1 Provide access to existing data bases

CASA's own catalog system was originally a Vms based relational data bases system containing several astrophysical catalogs, such as IRAS catalogs (point source, small scale structures, serendipitous survey, additional observations, low resolution spectra), the IUE Merged Log, SAO catalog, a cross index catalog (SAO/HD/GC/DM) and various smaller catalogs. Users could query the catalogs through "MCATS" (Multiple CATalogS, an application program developed at CASA), or directly through SQL (Structured Query Language) or RDO (DEC's proprietary query language).



λ	Mission
3456.67	IUE
12e+09	IRAS B1
4567	CCD 1
2345	CCD 2
3421	CCD 2
5674	CCD 4
23458	CCD 5

RA	DEC	Flux	λ
12h14'56'	-24°14'12"	8.945e+07	12μm
12h15'23'	-24°10'12"	8.945e+08	25μm
12h17'56'	-24°09'07"	4.565e+08	60μm

RA	DEC	Camera	λ
12h12'23'	2°23'57"	LWP	125
12h13'33'	6°23'06"	SWP	23
12h13'25'	2°24'01"	LWP	126
12h15'59'	5°29'57"	LWP	23
12h23'46'	9°09'07"	SWP	12
12h25'18'	6°18'23"	LWP	124
12h47'03'	1°59'45"	SWP	134

Relational databases

Mission description:

IRAS

Textual information

Spectra

Images

Figure 2: Images, spectra, tables or text describing astrophysical information.

The data bases named above included highly reduced information about data, but not the data itself. Data, such as IRAS skyflux images, or IUE spectra, were available locally at CASA in form of single files on hard disk, optical disk, or tape. Some of the database search functions would point to path- and filenames for data items when appropriate.

Parallel to the efforts of CASA, and on an immensely larger scale, NASA sponsored the development of ADS, the Astrophysical Data System, a remote network access system to data bases all across the astrophysical community.

Access to existing data bases and data base functions were made available through the CATALOG ACCESS menu in STAR. This included the use of MCATS as well as ADS, as long as these functions were supported on the workstation used to execute STAR.

2.2.2 Create new functionality for database selections

Information accessed through local or remote data bases must be visually inspected for verification of the goodness of quality or rejection and instigation of a new query. Because the visual inspection of data from data bases seemed an important aspect of the selection process, "quicklooks" of IRAS images and IUE spectra were made possible following a search through the appropriate catalogs. Quicklooks are replicas of the data at a reduced resolution, improving the scientist/machine dialogue by supporting a more discriminating selection of available data by the researchers. Information from astrophysical catalogs could be, if appropriate, overlaid on corresponding data sets.

2.3 Preprocessing and quantitative analysis

After raw data is acquired, either by new observations or from an existing archive, it must be subjected to a series of processing steps before it is suitable for visualization. We have divided the series of processing steps into two distinct classes: preprocessing and quantitative analysis. The division is based on where the major responsibility for software development historically has resided: preprocessing -- applying the necessary corrections and calibrations -- usually is the responsibility of the facility which originally acquired the raw data, the IUE Project in the case of IUE data for example; while quantitative analysis (measurements) -- extracting critical parameters from the corrected, calibrated data -- usually is the responsibility of the individual scientist, although in many cases a mission-specific facility will provide specialized tools to aid in the measurement of the primary data.

Preprocessing defines the major task of applying all known corrections and calibrations to the raw data in order to provide a final product that is as free as possible of systematic errors of instrumental origin, and which is presented in a readily usable form. An example is the geometrical correction, photometric linearization, and wavelength calibration of a ground-based CCD frame, or an IUE vidicon image. Preprocessing usually is the domain of the mission-specific processing center. However, it often is the case that the mission-specific center will release a data product that is tailored for the widest possible community of users, and is compatible with the (sometimes limited) capabilities of the mission processing system. There are many cases where sophisticated data enhancements techniques have been developed, often apart from the mission-specific center, which can significantly improve the quality of the data product. These enhancements typically are computationally intensive and can practically be applied to only small subsets of the overall data base. For the vast majority of scientific questions, the standard processing of the facility data is sufficient to provide the necessary answers. For a few applications, however, such data enhancement techniques are critical. In some cases, the necessary software is provided by the mission-center; in other cases, it is available only as custom-designed packages from the individual investigators.

Whereas the fundamental corrections to and calibrations of a particular primary data set usually are the exclusive domain of the production-processing team of a mission-specific center, the realm of quantitative analysis (measurements) is one more typically under the control of the individual scientist. In some cases, the scientist might make use of facility-developed software to expedite the measurement process; but in many cases the individual scientist will develop a specialized measurement approach that is carefully tailored to a specific research project. For example, one cool-star aficionado might desire the integrated flux of the Mg II k emission core in a particular star, whereas an interstellar-medium cognoscenti might desire to measure the strengths and radial velocities of the narrow absorption components in the k-line core of the same star. Although the scientific objectives might be quite different, and the object of attention - emission-line fluxes versus absorption-line parameters - might be quite dissimilar, the actual techniques by which one measures the relevant parameters might be essentially identical. Thus, a single generalized measurement tool - a Gaussian line fitting algorithm - might serve the purposes of a broad range of investigations based on widely differing data sets.

Expecting the scientist to choose from a variation of data sources during the first step of the data analysis process, we incorporated access to the following software packages into STAR: IUE/RDAF (IUE spectra reduction package), AIPS[2] (VLA radio map processing system), IRAF[3] (groundbased CCD reduction, as well as specialized processing modules for HST and ROSAT), and a series of inhouse-devel-

[2] Astronomical Image Processing System

[3] Image Reduction and Analysis Facility

oped image-enhancement procedures for the IRAS HCONs. Access to these packages provided mission-specific preprocessing tools as well as general measurement tools.

Because data formats varied with the use of different software packages, compatibility needed to be achieved between individual data items before multispectral analysis could be performed. STAR's solution to data transfer between software packages was to include import/export functions to transform between various data formats. Expecting that access to the above named packages would offer sufficient generalized preprocessing and measurement tools, only a few specialized tools were developed during the course of the project.

Newly developed tools included two semi-automatic preprocessing techniques for IRAS skyflux images that were designed and implemented as part of STAR. These preprocessing programs remove the background of IRAS skyflux images (zodiacal light) and reduce -- or remove -- the periodic stripes in skyflux data. The algorithms involved and results are explained and shown in the enclosed publication "Workstation-based Preprocessing of IRAS Sky-Flux Images", which is included as an appendix to this report. Before availability of the reprocessed skyflux data in 1992, these modules had a great value for visual and computational comparisons between different IRAS wavelengths bands.

New measurement tools were also include with some of the visualization functions, e.g. isosurface representation and flux measurements, as described in the next section.

2.4 Visual representation

2.4.1 Visual representation of data

Information extraction and preprocessing will produce corrected calibrated scientific data in highly-reduced form. The scientist now is confronted with a bulk of data for interpretation: data from individual missions; cross-correlated multi-wavelength data; and data in one or multiple dimensions. In our assumption of the importance of multi spectral data we stress the availability and use of information derived from different observation sources; consequently, however, the interpretation of results becomes more complex. The human mind is weak in making connections between large tables of numerical results. The expression "A picture is worth a thousand words" refers to the fact that the human visual system is capable of interpreting image data at an incredible faster rate than the same amount of data in tabular form. Therefore visualizing data, instead of simply reviewing large tables of numbers, will enhance the pace of data interpretation.

For example, it is much harder to interpret three single images, placed next to each other, than it is to interpret one color image incorporating these three images simultaneously. In practice one might choose a perceptive three dimensional color space, e.g. hue, lightness and saturation, that fits the principles of color perception of the human mind. Expressing multi-wavelength data characteristics by various colors at the same spatial coordinates results in a natural visual "format" for the scientist.

Other visualization techniques, besides color, also aid in the simultaneous presentation of multi-wavelength data sets: e.g. depth and animation. Both may add an additional dimension to spatial and brightness information in an image. In addition to sophisticated new techniques that aid in the process of simultaneous display of several layers of pictorial information, there are other familiar, fundamental visualization techniques that have been utilized in the past: two dimensional plots, pseudocolor displays, or graphic overlays on images and spectra.

2.4.2 Visualization concepts

The role of data visualization is to stimulate mental processes different from quantitative data analysis. Visual data analysis offers an overview of data characteristics through browsing, often leading to an intuitive understanding of data characteristics and their relationships by sacrificing accuracy in interpreting the data values. Because the human visual system emphasizes spatial relationships, up to three data characteristics can be represented in a natural, intuitive way in form of spatial dimensions. Data visualization is an indirect way of interpreting data: instead of being interpreted from their natural, usually quantitative characteristics, data are first encoded into a pictorial representation. The encoding process bears the danger of creating artifacts and therefore missing the correct interpretation: e.g. abrupt color changes may mislead by pointing to discontinuities in a data set; subjective assessments of patterns may lead to other misinterpretations.

A visual representation of data values should take into account the data characteristics as well as the interpretation intent of a scientist (Mackinlay, 1986; Wehrend and Lewis, 1990; Robertson, 1990) to encode data values into a pictorial representation from which a scientific interpretation can be derived (shown in Figure 3).

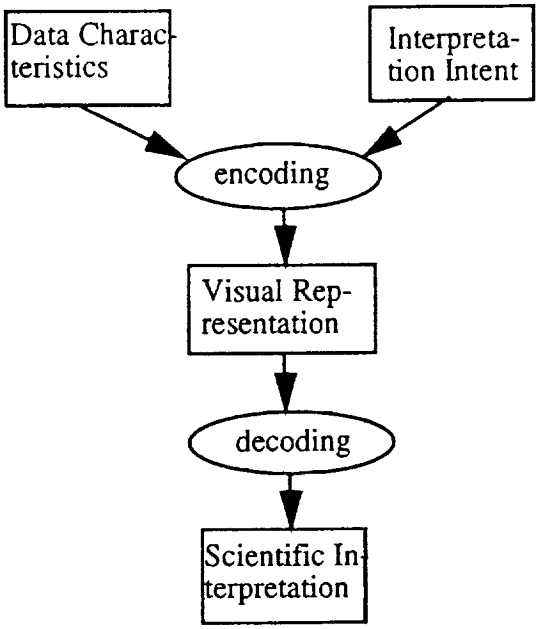


Figure 3: Process to encode data values into a visual representation.

2.4.3 Visualization tools developed for STAR

IDL[4], the software platform chosen for the development of STAR, incorporated many basic and higher-level visualization techniques (e.g. display of one and two dimensional data sets; pseudocoloring; perspective projections). Over the past years many visualization techniques that we had planned to implement, or started to implement, have recently become available, specifically in the area of three dimensional display techniques, e.g. isosurfaces or data slicers.

During our first project years we found a high demand for simple visualization tools that allowed interactive manipulation as compared to demands for complex visualization tools. Data slicers and 3-d representations of structures did not get much attention by CASA's scientists in the early stages of our inquiries. As a consequence, in the beginnings of the project, design and development of visualization tools concentrated on these early needs. As can be seen in chapter 3, latter demands were for highly dimensional and more complex visualization techniques.

Therefore simple visualization techniques, such as profiling objects, color table editors and switch boards, interactive data type conversions (the dynamic range of some astrophysical data, e.g. IRAS data, is very high compared to the 256 colors/gray values visible on a standard graphics workstation), were included into early versions of STAR. Later more complex visualization techniques followed,

[4] Interactive Data Language by Research Systems, Inc.

such as interactive color coding techniques (e.g. HLS[5], HSV[6], 24-bit to 8-bit color compression), volumetric data displays (e.g. iso surface display) and data slicers.

2.4.4 Feasibility study with AVS[7] and IDL

In addition to developing new tools we studied the feasibility of well known state-of-the-art commercial visualization packages (AVS and IDL) for astrophysical data analysis. Results of these experiments are included in the example section. While we found AVS to be a sophisticated (and easy to use) graphics package, it would not fit into the astrophysical environment in its current status. Some of the shortcomings we found with AVS were the lack of its use for quantitative analysis. While AVS nicely encoded data fields into pictures, e.g. showing contour lines of a 2-d array of data values, it was often impossible to retrieve the original numbers from the visual representations. The lack of data formats appropriate for the astronomical community as well as the relatively high licence fees furthermore restricted the use of AVS for astrophysical environments.

IDL has long found its place in astrophysical data analysis. Therefore a fair amount of visualizations already allow interactive, quantitative analysis. In the case of the “data slicer” and the “iso surface” displays, we have expanded the original source code to permit direct measurements and computations on the surface of arbitrary slices and inside isosurfaces. These new visualization/analysis tools are described further in the appendix.

2.4.5 3-d Interaction devices

With the exploration of volumetric data, the problem of how to interact with 3-dimensional data on a 2-d screen became obvious. Input devices (e.g. mouse) as well as output devices (e.g. 8-bit color workstation displays) that are currently available at reasonable prices are limiting the scientist.

An example might illustrate the problem in a better way: A scientist looks at an isosurface display of an astrophysical data cube, as pictured in Figure 18. By rotating the cube and its content, the 2-d screen clearly conveys the effect of a 3-d object. In order to analyze part of the cube in more detail, the scientist wishes to extract a subcube. At this moment several problems hamper the interaction:

- In order to “reach into” the cube (spatial positioning), the rotating movement has to be stopped. This results in collapsing the 3-d information into a static 2-

[5] Hue-Lightness-Saturation

[6] Hue-Saturation-Value

[7] Application Visualization System by AVS, Inc.

d perspective projection. An output device such as a stereo-screen coupled with stereo glasses or a virtual environment would solve this problem.

- A mouse is moved over a 2-d surface to simulate 3-d movement in various ways: e.g. any 2-d coordinate plane in 3-d space can be selected consecutively to reach any spatial position in 3-d space. While 3-d spatial feedback is already poor on the 2-d screen, the separate movement is also non-intuitive to a natural gesture of grasping an object.

In view of the lack of 3-d input and output devices that are effective as well as cheap, we performed a short study on the availability, effectiveness and costs of such devices. The result of this study is included in the appendix of this report.

2.5 User interface design

In order to build a system that is liked and used by researchers, we interviewed our potential users on their likes and dislikes of analysis systems, specifically noting aspects of the user interface (Mickus, 1991; see section 6.3). The breadth of answers lends itself to a discussion that goes beyond the scope of this report. To summarize, common wishes were:

- support major application systems, such as IRAF, IUE/RDAF, IDL, AIPS;
- use on-line documentation of software;
- use windows, widgets and interactive tools;
- offer a dynamic system, e.g. easily expandable if new software is to be included, or when new application packages are to be installed.

STAR's user interface therefore incorporated all these aspects:

- In its start-up state the system allows access to IRAF, IUE/RDAF, IDL, AIPS. Customized startup files (e.g. for IRAF and IDL) can be created at one time and used for future purposes.
- On-line help and documentation is available on application software packages developed in-house. In order to keep documentation of user-built modules as updated as possible, we provided an automatic documentation tool, which extracts comments from the source code to make them available as source code documentation. Documentation for external software packages is the responsibility of the developers.
- STAR is built in a workstation environment, on top of X-windows and IDL (later versions on IDL/widgets) making therefore extensive use of windows, widgets and interactive tools.

Feedback from scientists on the use of STAR was solicited throughout the design and development period. New research results in the area of Human/Computer Interaction (HCI) were employed to encourage feedback about the scientist's desires. We were able to take advantage of a strong HCI research environment available at the University of Colorado. Under consultancy of Dr. Lewis, cognitive design techniques were used throughout the first 1.5 project years to solicit feedback from CASA's scientists about the user interface design and visualization tools.

Besides soliciting feedback at CASA, we also solicited feedback from a larger audience outside the University of Colorado about the design goals of STAR (see also section 6.5). New aspects and desires of these audiences further influenced the design of STAR.

A detailed review of design issues concerning the data analysis environment is given in the enclosed publication "Design and Development of a Data Visualization System in a Workstation Environment", which is attached as an appendix to this report.

3. Examples of Work in the Project

3.1 Data analysis cycle and user interface

Figure 4 and 5 show in a schematic and actual view the main user interface of STAR. Pull-down menus reveal the highest level of functions to perform

DATA INPUT/OUTPUT	Read in / write out data to tape or disk; conversions between various data formats; quick saving and restoring of data variables
CATALOG ACCESS	Retrieve information from databases by accessing databases/catalogs directly or by executing database programs
PREPROCESSING	Apply necessary corrections and calibrations to data
ANALYSIS	Extract, measure and mark objects
VISUALIZATION	Convert data to visual representations

Square buttons denote external software packages that may be executed by clicking on the corresponding button. Round buttons reveal different menu functions for CCD, IUE or IRAS based analysis. A "PROBLEM" button connects the user to STAR's software developers to leave complaints, demands or recommendations. A status window reports the current status of the software system, e.g. "Error"; "Waiting for user input..."; "Computing...".

Figure 6 shows one submenu of the "CATALOG ACCESS" button: MCATS, CASA's local Vms catalog access program, offers access to locally stored astrophysical catalogs.

Figure 7 shows another "CATALOG ACCESS" function: "Quicklook" displays data files selected through database programs in reduced resolution on the screen.

Figure 8 shows a "PREPROCESSING" function to flatten IRAS skyflux images. The upper left image shows the original image, the lower right the result after flattening.

Figure 9 shows the interactive measurement of fluxes and positions in the image by moving mouse/cursor over image pixels. Corresponding horizontal and vertical profiles are plotted when clicking the mouse button.

Figure 10 shows an axonometric view of an IRAS skyflux image combined with its contour lines.

3.2 Destriping and flattening of IRAS skyflux images

Two preprocessing techniques to flatten and destripe IRAS skyflux images were designed, documented and implemented in order to enable visual and computational comparisons between different IRAS wavelength bands. The flattening process (removal of zodiacal light) is shown in Figure 8. Several examples and detailed explanations are given in Appendix A.

3.3 Simple visualization techniques

Figure 11 shows an interactive surface plot of flux values. The control window to the right defines the current view point position.

Figure 12 visually correlates three of the four IRAS skyflux bands shown in the upper picture and displays the result as a color picture on an I^2S/IVAS 24-bit color image display station. A similar color coding technique is used to correlate three IRAS skyflux bands on a (much cheaper) 8-bit color display in Figure 13. The difference in the appearance of the pictures relates to the use of different input images in the encoding algorithm.

An interactive switch board (Figure 14) facilitates the use of available color tables.

A high dynamic range of flux values (represented by the y-axis in window

"STRETCH") is being converted interactively to the available color range of the graphics workstation (Figure 15).

Figure 16 shows a flexible adjustment tool of the internal color lookup table. Linear as well as non-linear transformations can be defined interactively. Besides user-defined conversions between data values and display values, predefined transformations, e.g. statistical stretches, are available.

Figure 17 shows the interactive zooming tool.

3.4 Data cube visualizations

Collaboration with Dr. John Bally at CASA started beginning of 1992 and gave us an opportunity to present new visualization techniques as recently developed (or still under development) by the computer graphics community. Following is a short explanation of Bally's data sets and scientific goals in order to better understand the visual representations chosen.

Data is collected by a 7 m telescope dish owned by ATT Bell Labs in New Jersey. It operates at a frequency between 23 to 43 Ghz, corresponding to a wavelength of 1.3 cm to .7 cm. The collected data is in form of 2-d image tiles for each measured frequency. Processing of the collected raw data values from the heterodyne receiver results in even gridded data values defined in three dimensions (spatial, spatial, frequency). The data values correspond to a count of carbon monoxide molecules at that specific spatial location and frequency. Data values may range between -32000 and +32000.

Carbon monoxide is used to trace molecular clouds. It is important to understand changes of the molecular cloud in space as well as in frequency. Therefore, scientific tasks in interpreting data values are: observe, if the cloud is expanding; look for indications that the cloud is collapsing; in what direction is it moving; identify dense matter at each stage of frequency.

It is important to express essential data characteristics in the resulting visual representations. In the case of the astrophysical data cubes, such essential characteristics are spatial location as well as frequency and the data value itself. Leaving both spatial dimensions in their natural form and mapping frequency into a third spatial dimension created an even gridded cube ("data cube") with the data values expressed as voxels.

However, the various slices of spatial data values could also collapse into one single slice, where spatial dimensions are represented in their natural form, but various data values along one frequency dimension are expressed in one pictorial "glyph".

It is important to represent the data is an effective way, so that the decoding process from pictures to scientific interpretation is quick and accurate. The following visual representations were chosen and discussed with Dr. John Bally:

3.4.1 Iso surfaces

Data values of a certain threshold were connected to create iso surfaces. This is a well known rendering technique of a data cube representation. In this representation, the overall shape of the data can be observed as well as isolated volumes (see Figure 18). Understanding the overall distribution of the carbon monoxide in the given spatial-spectral dimensions is important in order to understand the detailed quantitative information. The iso surface representation can be enhanced by adding individual slices through the data cube (see Figure 19) or by using several transparent iso surfaces (Figures 20).

3.4.2 Translucent representations

Rays penetrate the data cube from a chosen point-of-view and accumulate values of opacity related to corresponding data values. This representation inflicts a transparent characteristic on the molecular clouds, very much like the visual form of real clouds. It allows to look into the cloud as opposed to observing the surface only. Because the scientist felt a natural understanding of this representation, it was favored as compared to any other representation.

Figure 21 shows a translucent rendering of the cube by looking at the data from one side: one spatial dimension increases to the right, the frequency increases from bottom up. The rapid changes of the data values in the mid-frequencies show special characteristics of carbon monoxide at these frequencies. Figure 22 shows the same data cube using the same representation looking from top down onto the cube.

Figure 23 uses colors relevant to the frequency content: high frequencies are colored blue, low frequencies are colored red. During rotation of the display, the viewer will therefore be constantly aware of the frequency range s/e is looking at.

3.4.3 Data slicer

To monitor the change of one data value in relation to its neighbor values, a data slicer was used. Even though a data slicer can only monitor the neighbors surrounding a certain data value inside a plane, flexibility in placing the slices inside the cube can monitor various changes. Figure 24 shows four slices cutting through the cube parallel to the x/y plane, enhancing the understanding of the movement of the cloud through frequency. Figure 25 shows three orthogonal slices through the data, intersecting in the center of the cube. Figure 26 shows an arbitrary slice through a data cube.

3.4.4 Glyphs

To collapse all (or a subset of) data slices along the frequency dimension into one single two dimensional image, one must map all data values along one frequency dimension into one complex "glyph". The difference of one glyph from its neighboring glyph relates to the spectral characteristics of carbon monoxide and can be interpreted accordingly.

The resulting image can also be seen as one entity, therefore allowing interpretation of the overall distribution and change of carbon monoxide in the data cube. Visual representations of collapsed data slices leave it up to the human visual system to decide if the focus is on large-scale or small-scale structures.

Figure 27 shows a glyph representation of nine consecutive slices: color slices are used inside each red square to indicate various spectral responses at each spatial location. Figure 28 encodes five slices into five characteristics of a cube: width, height, depth, color and view point.

3.5 Expansions to commercial visualization software

One major point of dissatisfaction for Dr. Bally was the fact that in many cases of commercial visualization software no interaction could take place on the visual display itself. E.g. after displaying isosurfaces of a data cube, availability for quantitative analysis, such as integrating flux values inside an isosurface, is very important to interpret data. We therefore modified the IDL modules of "data slicer" and "isosurfaces" to include quantitative analysis.

Figure 29 shows three isosurfaces and a corresponding count of flux values inside these isosurfaces.

Figure 30 shows a grid in an arbitrary plane through an isosurface. The grid aids in probing data values inside the data cube in relation to the isosurface.

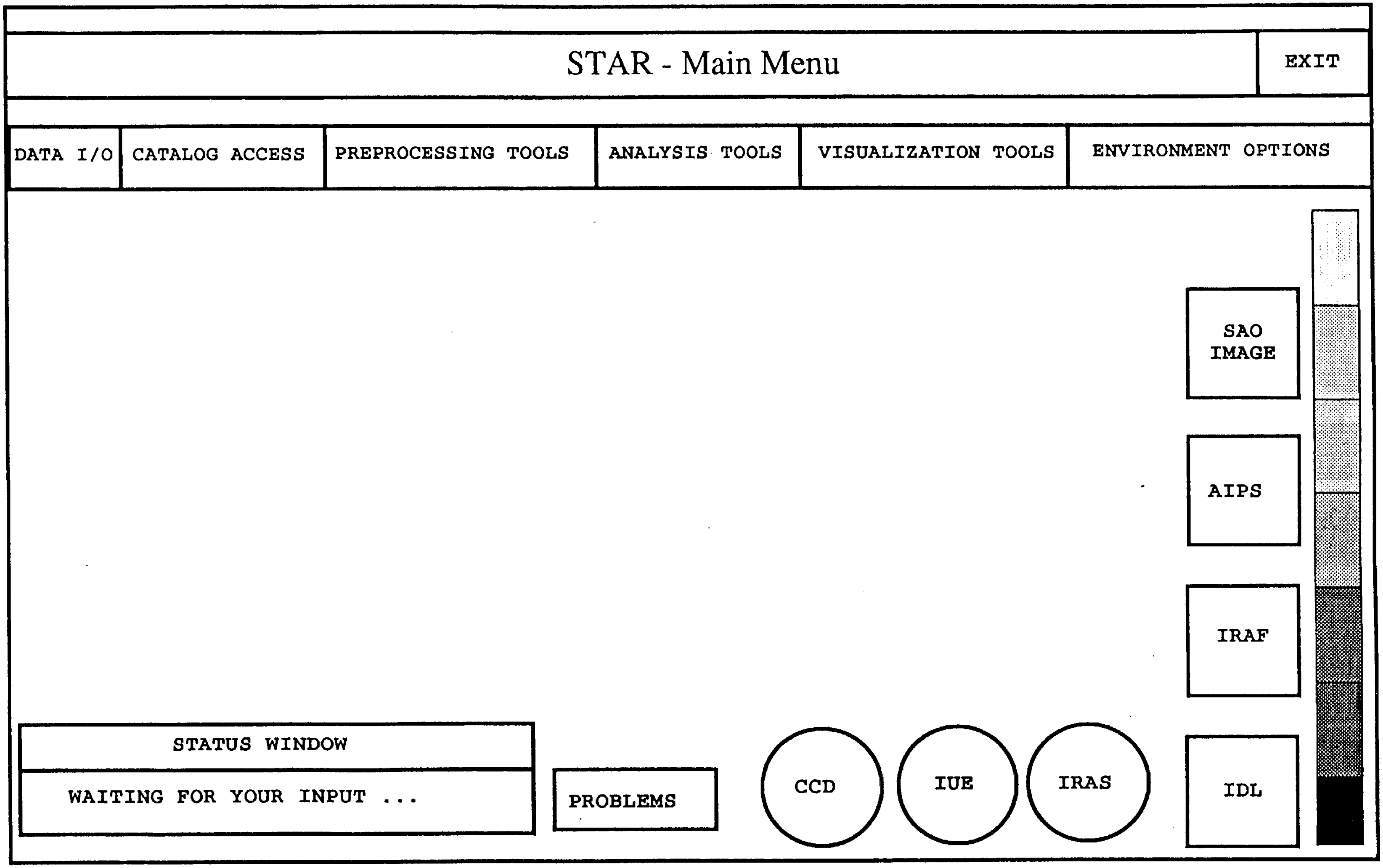
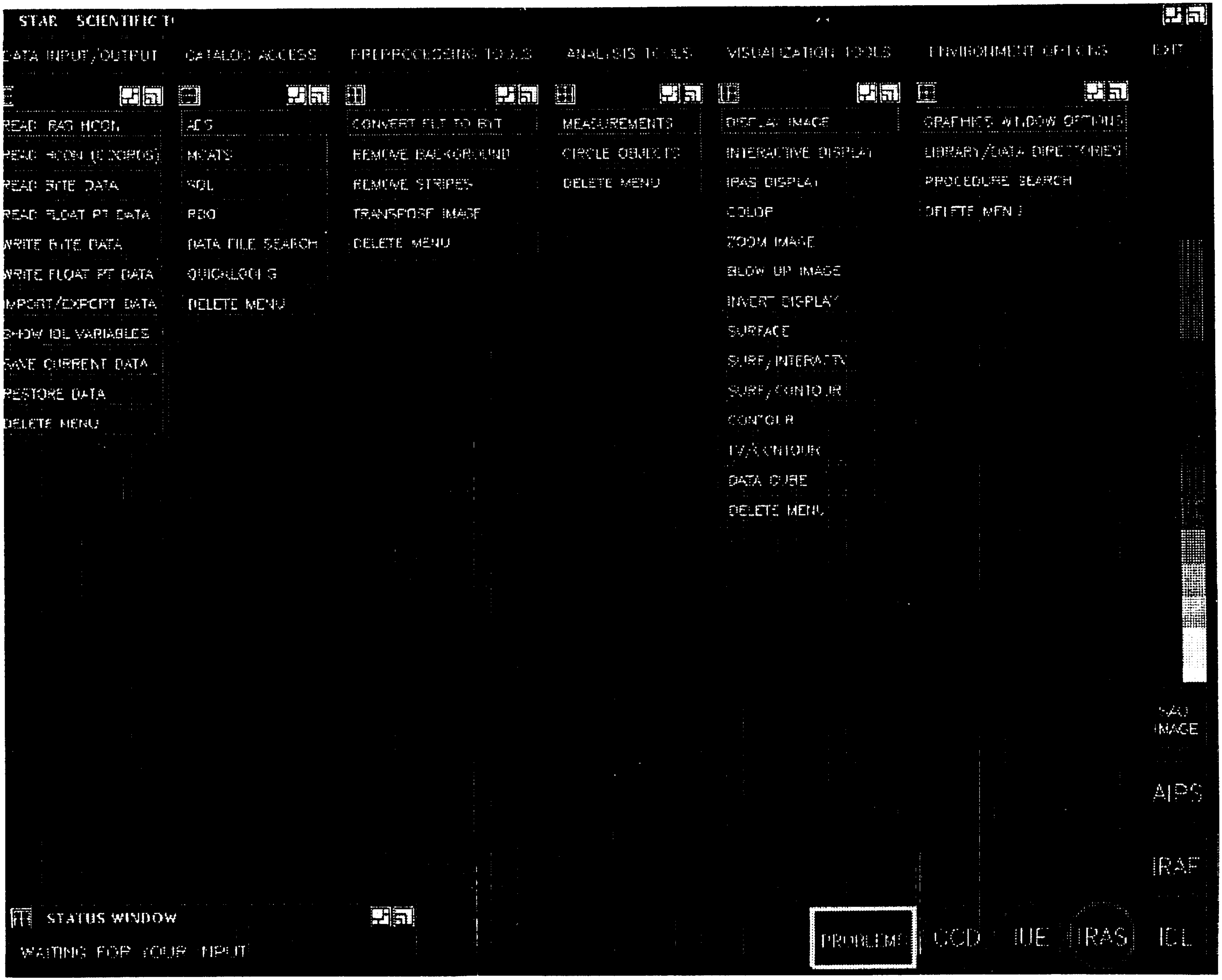
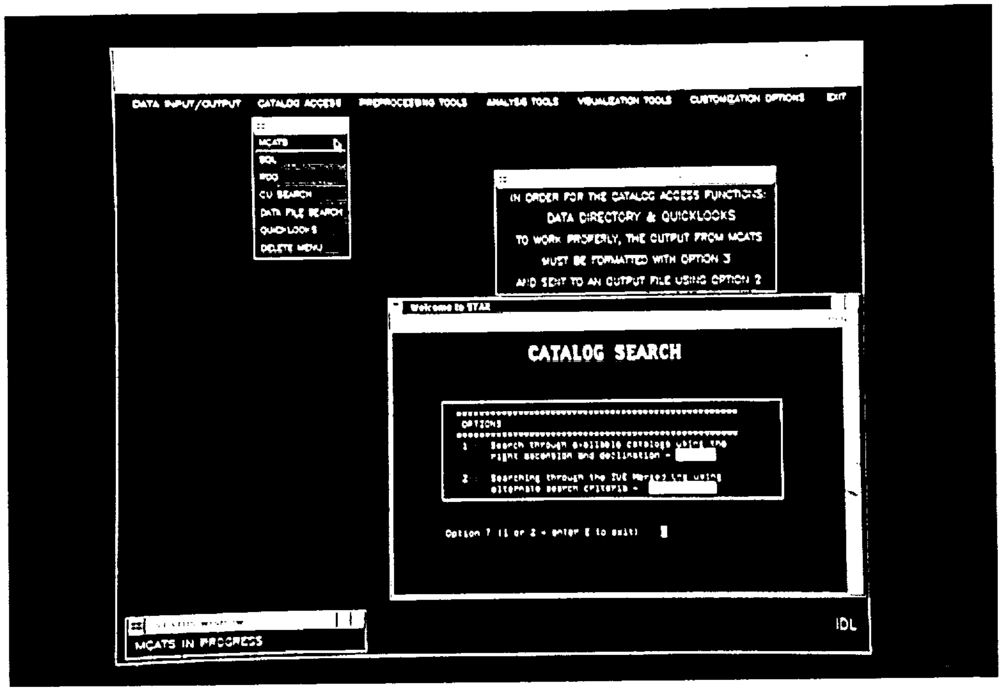


Figure 4: Schematic view of the main user interface of STAR.

Figure 5: Actual view of the main user interface of STAR.



WAITING FOR YOUR INPUT



IN ORDER FOR THE CATALOG ACCESS FUNCTIONS:
DATA DIRECTORY & QUICKLOOKS
TO WORK PROPERLY, THE OUTPUT FROM MCATS
MUST BE FORMATTED WITH OPTION 3
AND SENT TO AN OUTPUT FILE USING OPTION 2

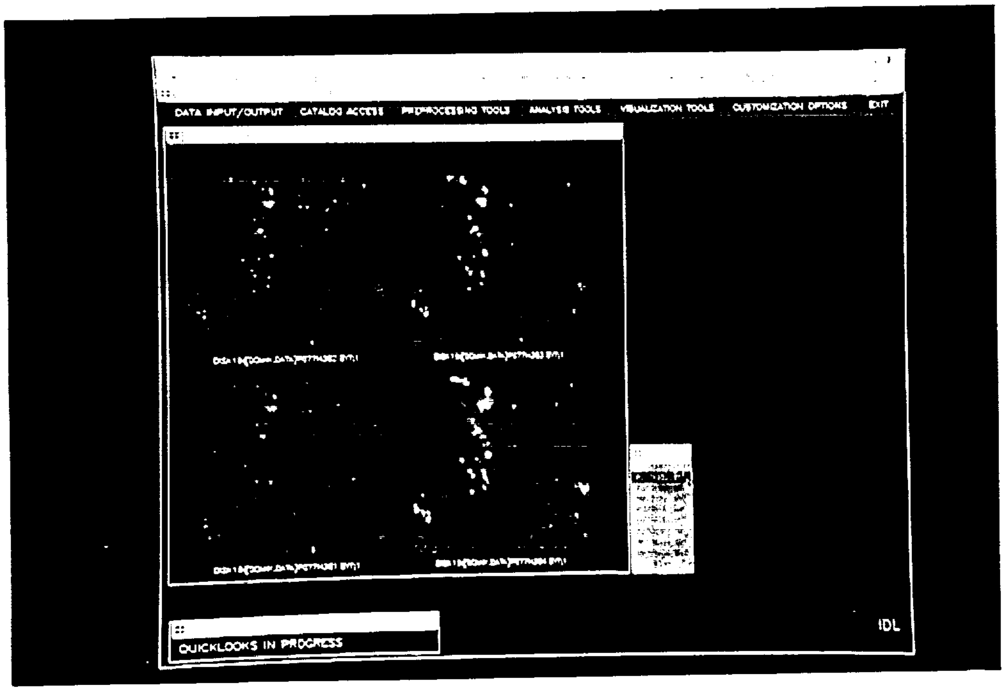
CATALOG SEARCH

```
OPTIONS
1    Search through available catalogs using the
     right ascension and declination -
2    Searching through the IUE Merged Log using
     alternate search criteria -

Option ? (1 or 2 - enter E to exit)
```

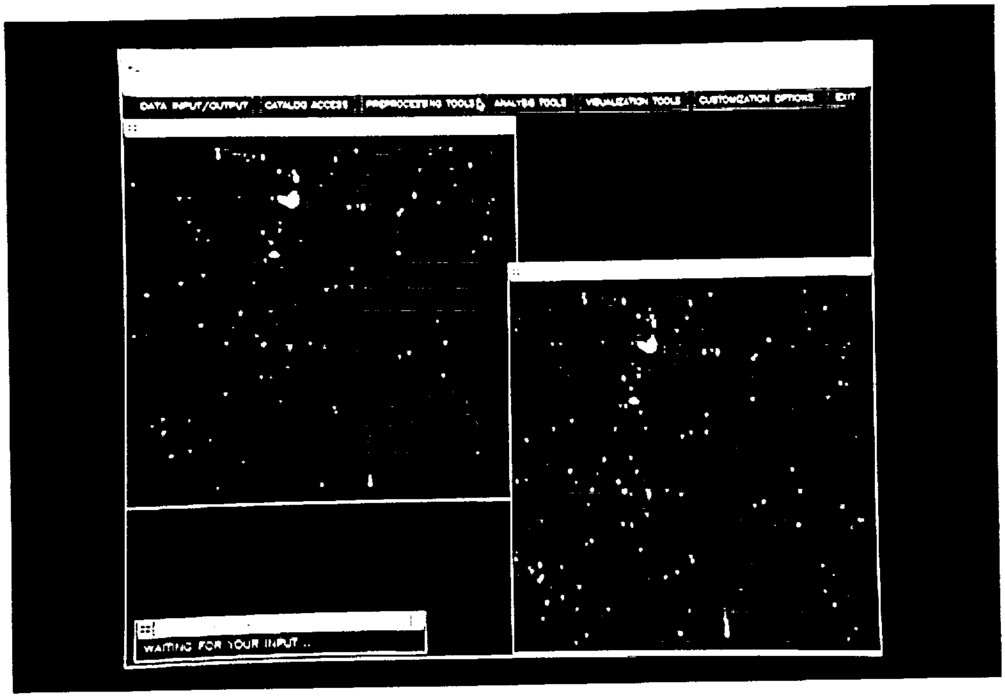
MCATS IN PROGRESS

Figure 6: Submenu of "CATALOG ACCESS" button: MCATS, CASA's local catalog access program, offers access to locally stored astrophysical catalogs.



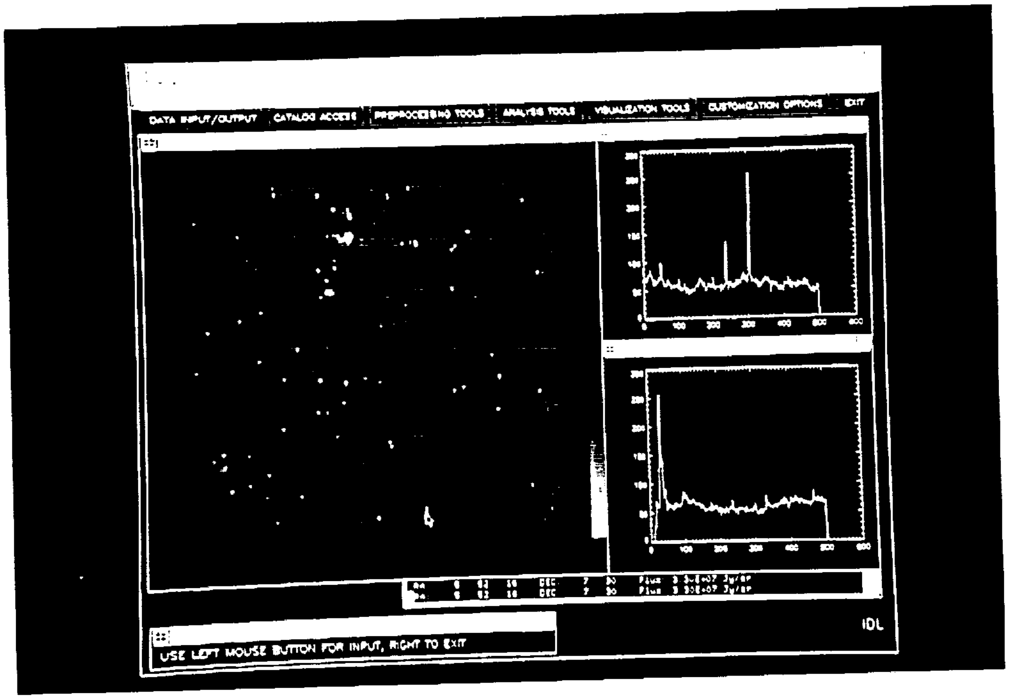
QUICKLOOKS IN PROGRESS

Figure 7: Another "CATALOG ACCESS" function: "Quicklook" displays data files selected through database functions in reduced resolution on the screen.



WAITING FOR YOUR INPUT ...

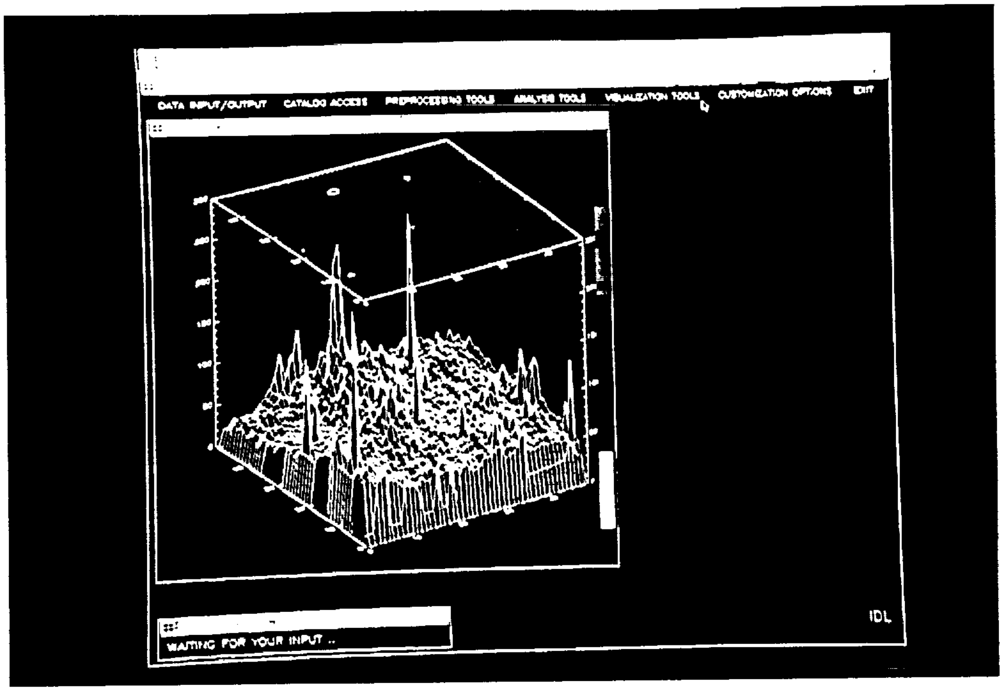
Figure 8: "PREPROCESSING" function to flatten IRAS skyflux images. The upper left image shows the original image, the lower right the result after flattening.



RA 6 52 16 DEC 7 30 Flux [illegible] Jy/SR
RA 6 52 18 DEC 7 30 Flux [illegible] Jy/SR

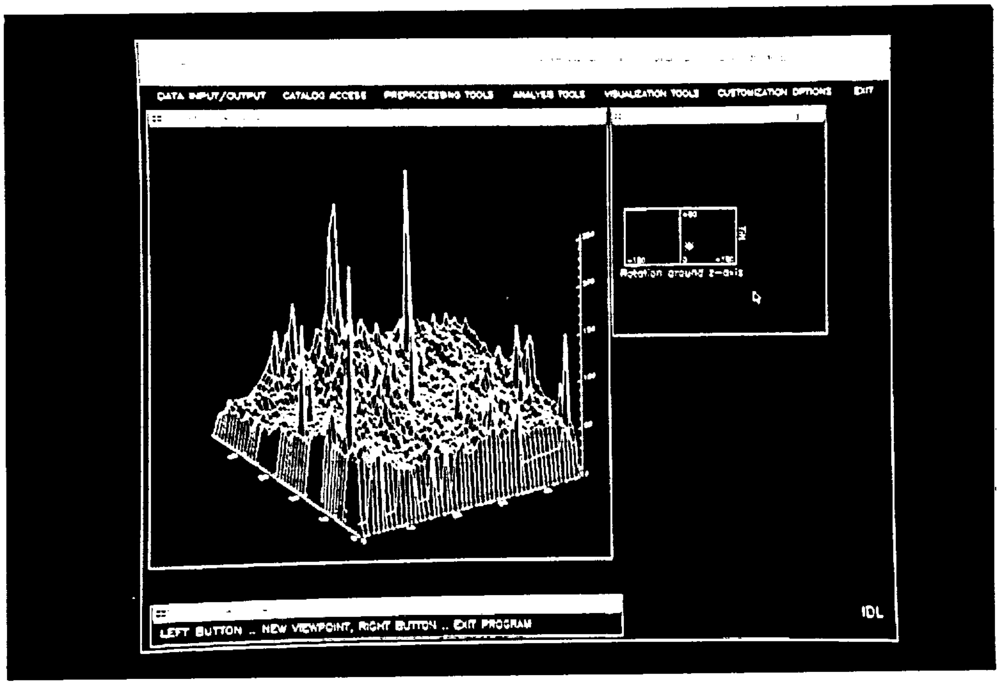
USE LEFT MOUSE BUTTON FOR INPUT, RIGHT TO EXIT

Figure 9: Interactive measurement of fluxes and positions in the image by moving mouse/cursor over image pixels. Corresponding horizontal and vertical profiles are plotted when clicking the mouse button.



WAITING FOR YOUR INPUT ..

Figure 10: Axonometric view of an IRAS skyflux image combined with its contour lines.



LEFT BUTTON .. NEW VIEWPOINT, RIGHT BUTTON .. EXIT PROGRAM

Figure 11: Interactive surface plot display of flux values. The control window to the right defines the current view point position.

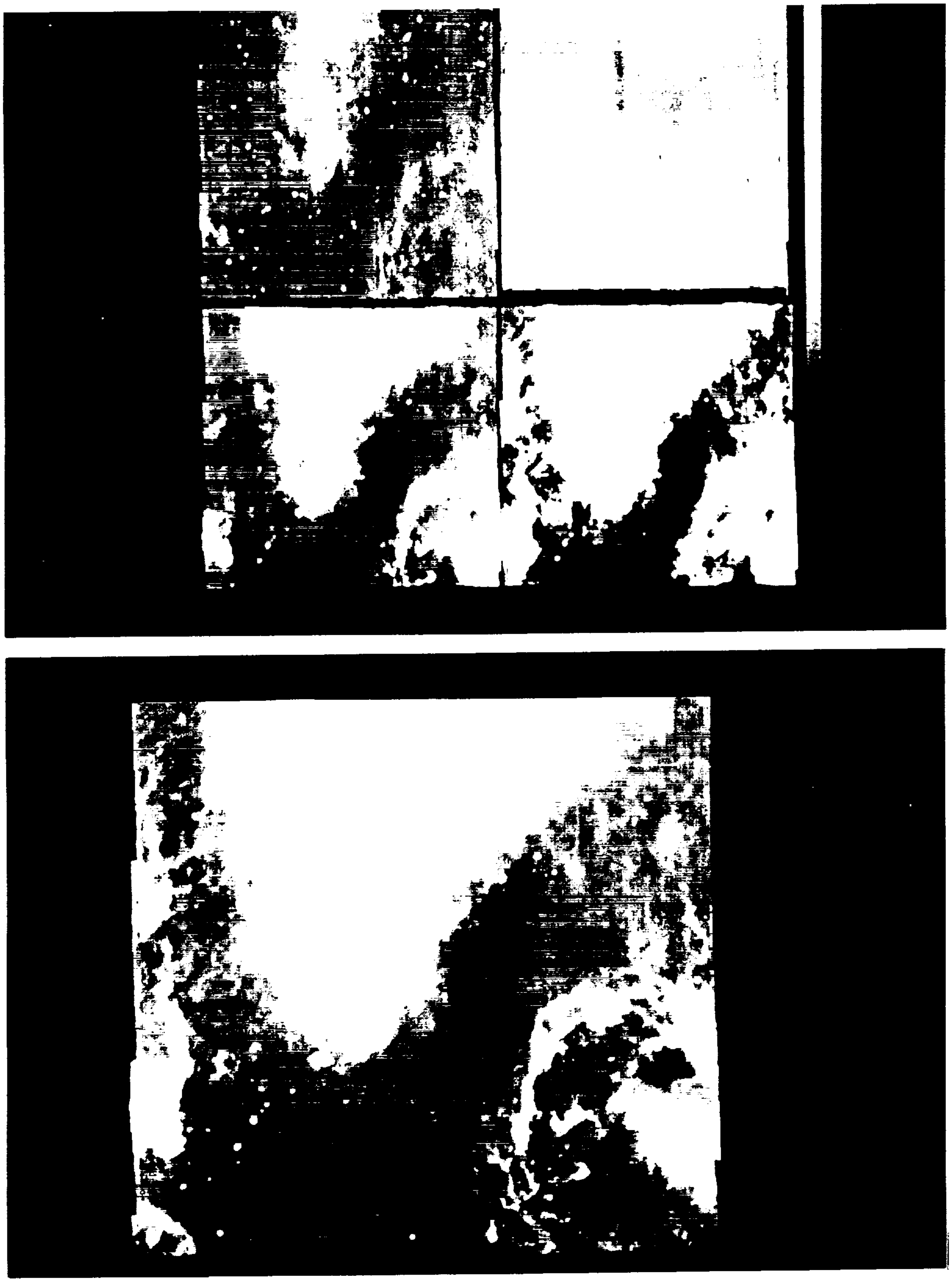


Figure 12: Visually correlates three of the four IRAS skyflux bands in the upper picture and displays the result as a color picture on an I^2S/IVAS 24-bit color image display station.

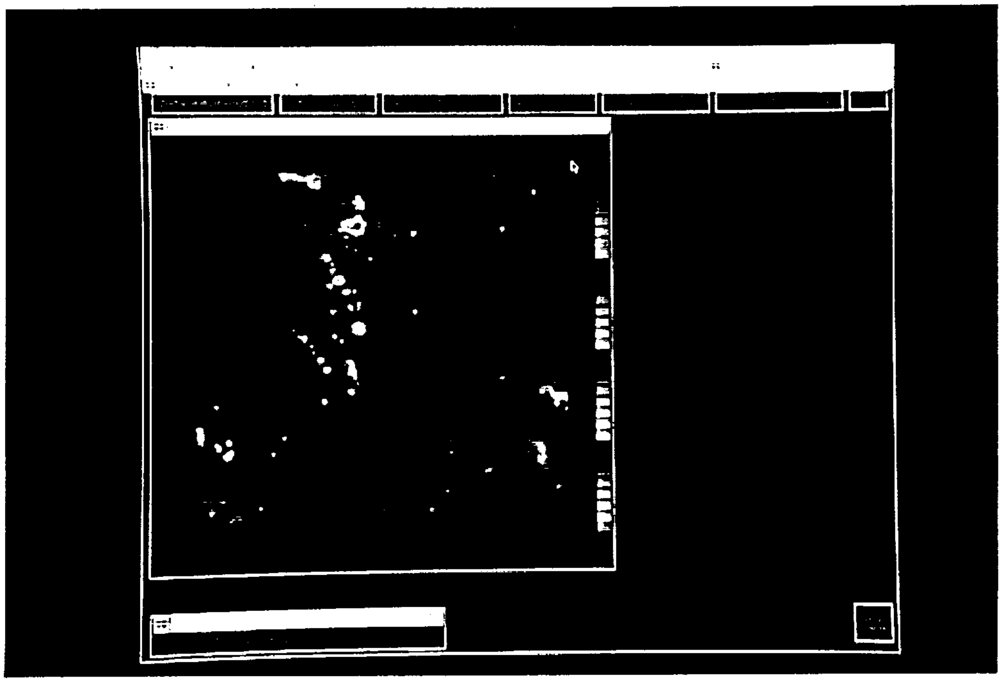


Figure 13: Visually correlates three of four IRAS skyflux bands (see Figure 12) and displays the result as a color picture on the monitor of an 8-bit graphics display.

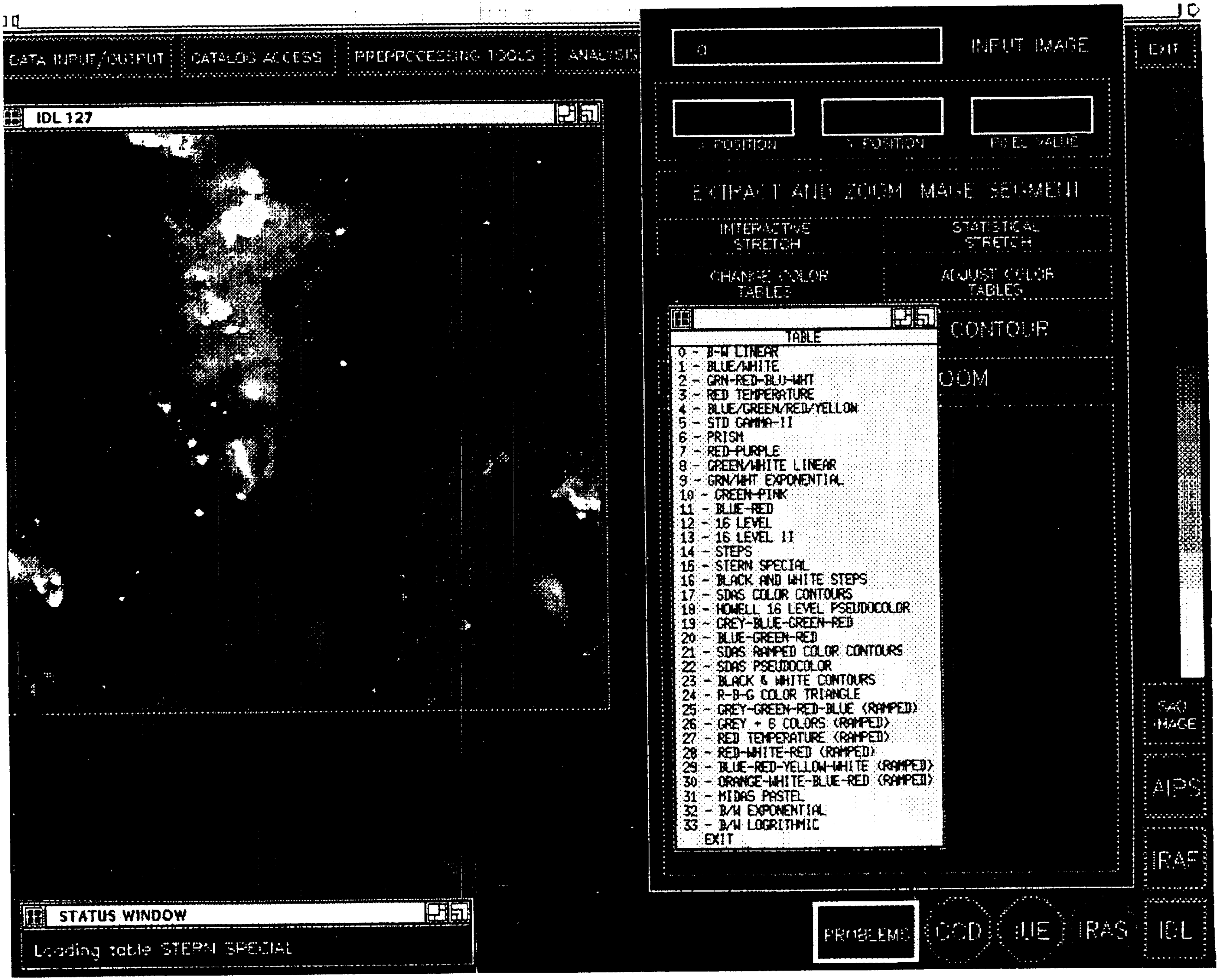


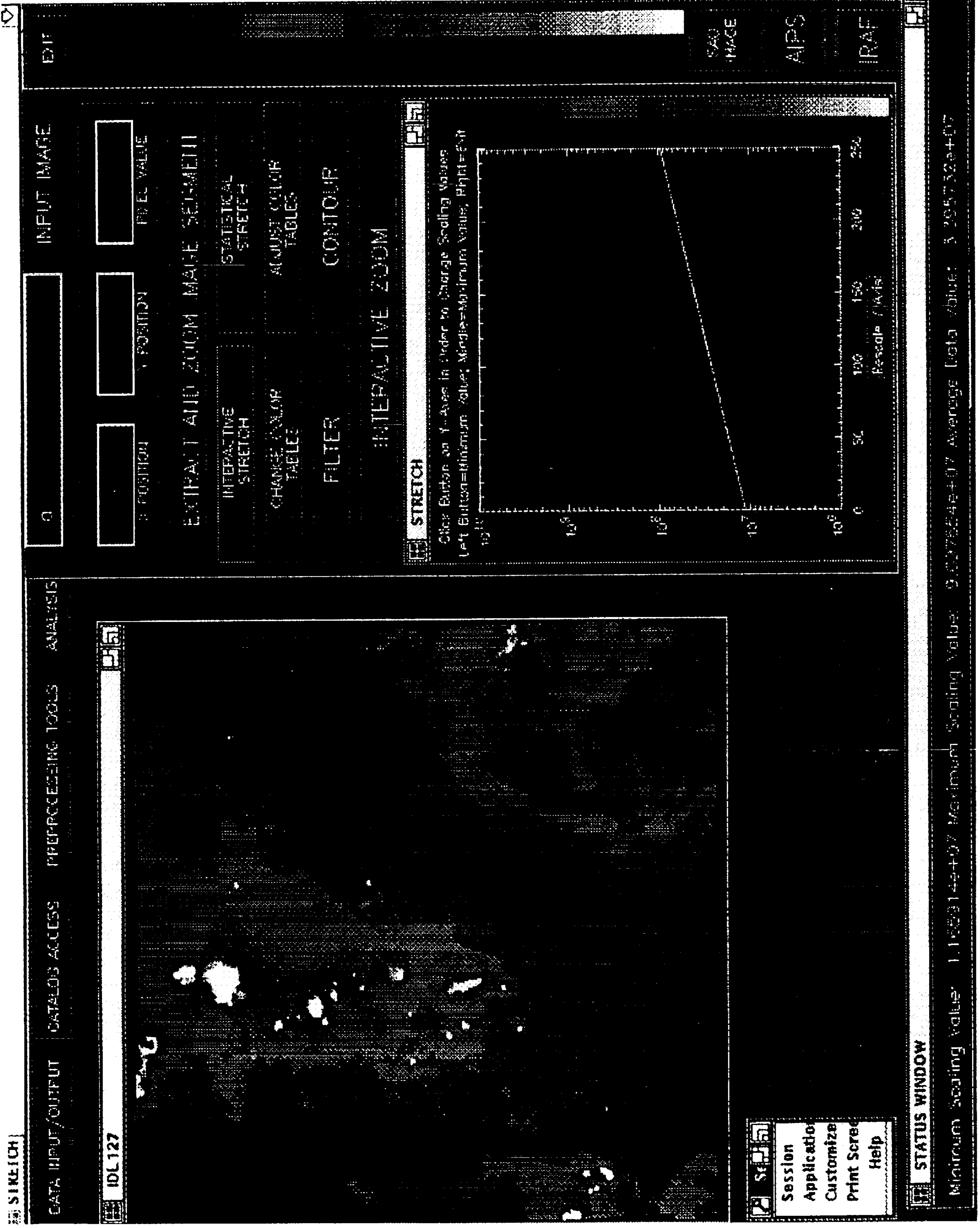
TABLE
0 - B-W LINEAR
1 - BLUE/WHITE
2 - GRN-RED-BLU-WHT
3 - RED TEMPERATURE
4 - BLUE/GREEN/RED/YELLOW
5 - STD GAMMA-II
6 - PRISM
7 - RED-PURPLE
8 - GREEN/WHITE LINEAR
9 - GRN/WHT EXPONENTIAL
10 - GREEN-PINK
11 - BLUE-RED
12 - 16 LEVEL
13 - 16 LEVEL II
14 - STEPS
15 - STERN SPECIAL
16 - BLACK AND WHITE STEPS
17 - SDAS COLOR CONTOURS
18 - HOWELL 16 LEVEL PSEUDOCOLOR
19 - GREY-BLUE-GREEN-RED
20 - BLUE-GREEN-RED
21 - SDAS RAMPED COLOR CONTOURS
22 - SDAS PSEUDOCOLOR
23 - BLACK & WHITE CONTOURS
24 - R-B-G COLOR TRIANGLE
25 - GREY-GREEN-RED-BLUE (RAMPED)
26 - GREY + 6 COLORS (RAMPED)
27 - RED TEMPERATURE (RAMPED)
28 - RED-WHITE-RED (RAMPED)
29 - BLUE-RED-YELLOW-WHITE (RAMPED)
30 - ORANGE-WHITE-BLUE-RED (RAMPED)
31 - MIDAS PASTEL
32 - B/W EXPONENTIAL
33 - B/W LOGRITHMIC
EXIT

STATUS WINDOW
Loading table STERN SPECIAL

Figure 14: An interactive switch board facilitates the use of available color tables.

ORIGINAL PAGE IS
OF POOR QUALITY

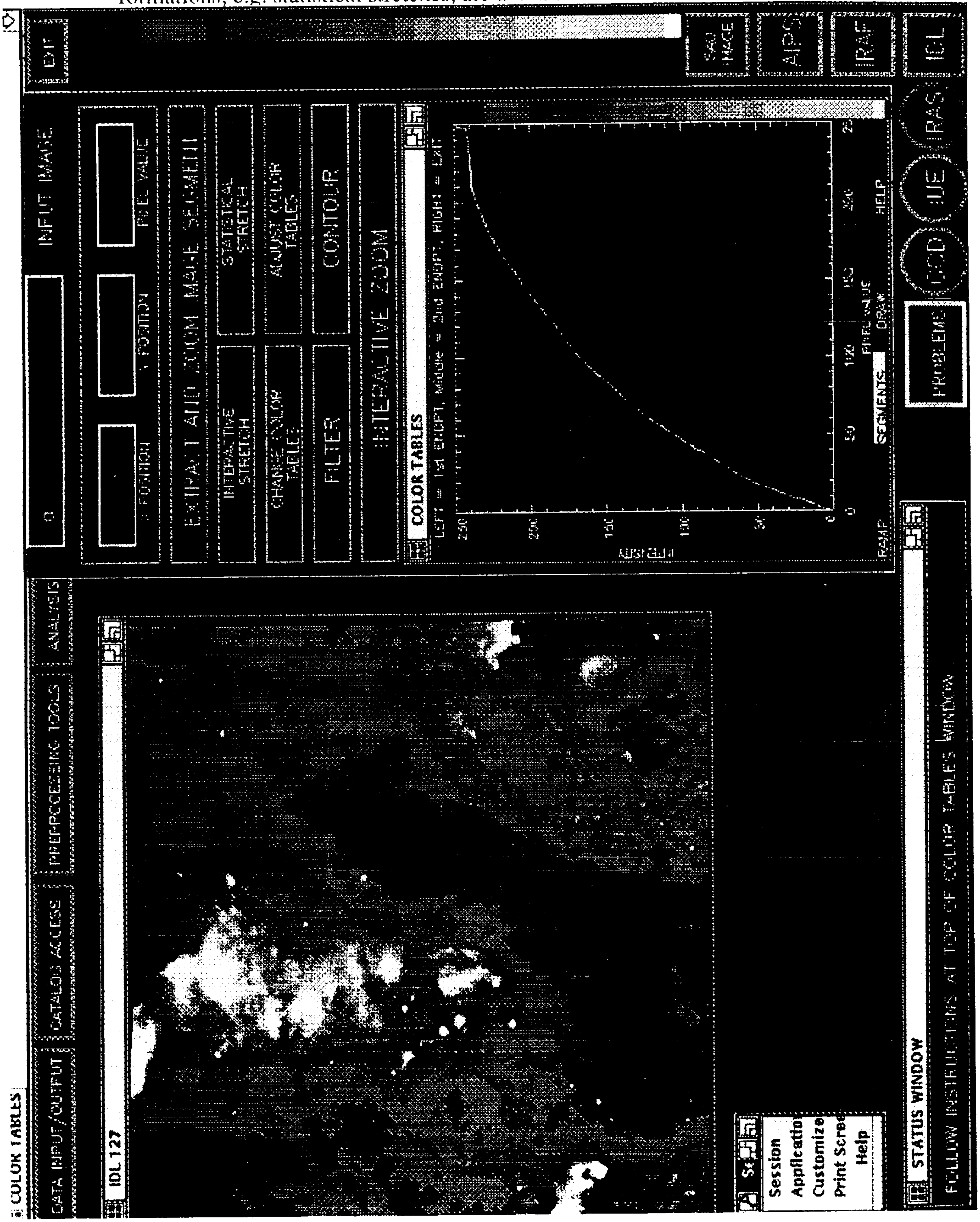
Figure 15: A high dynamic range of flux values (y-axis in window "STRETCH") is being converted interactively to the available color range of the graphics workstation.



Click Button on Y-Axis in Order to Change Scaling Values
Left Button=Minimum Value; Middle=Maximum Value; Right=Exit

Minimum Scaling Value: 1.1592143e+07 Maximum Scaling Value: 9.0278524e+07 Average Data Value: 3.2957132e+07

Figure 16: Adjustment tool of the internal color lookup table. Linear as well as non-linear transformations can be defined interactively. Besides user-defined conversions between data values and display values, predefined transformations, e.g. statistical stretches, are available.

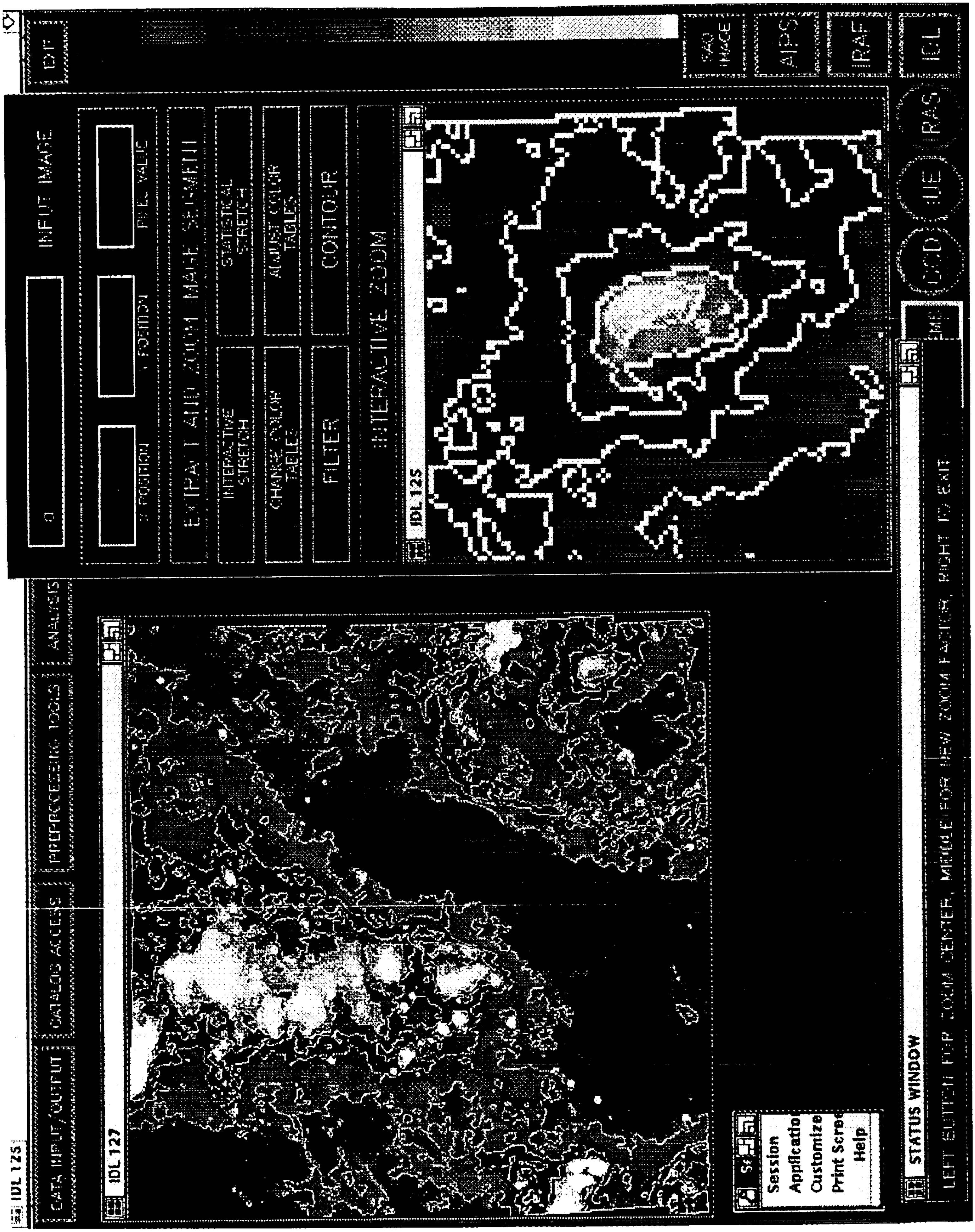


LEFT = 1st ENDPT, Middle = 2nd ENDPT, RIGHT = EXIT

FOLLOW INSTRUCTIONS AT TOP OF COLOR TABLES WINDOW

Figure 17: Interactive zooming and contouring tool.

ORIGINAL PAGE IS
OF POOR QUALITY



LEFT BUTTON FOR ZOOM CENTER, MIDDLE FOR NEW ZOOM FACTOR, RIGHT TO EXIT

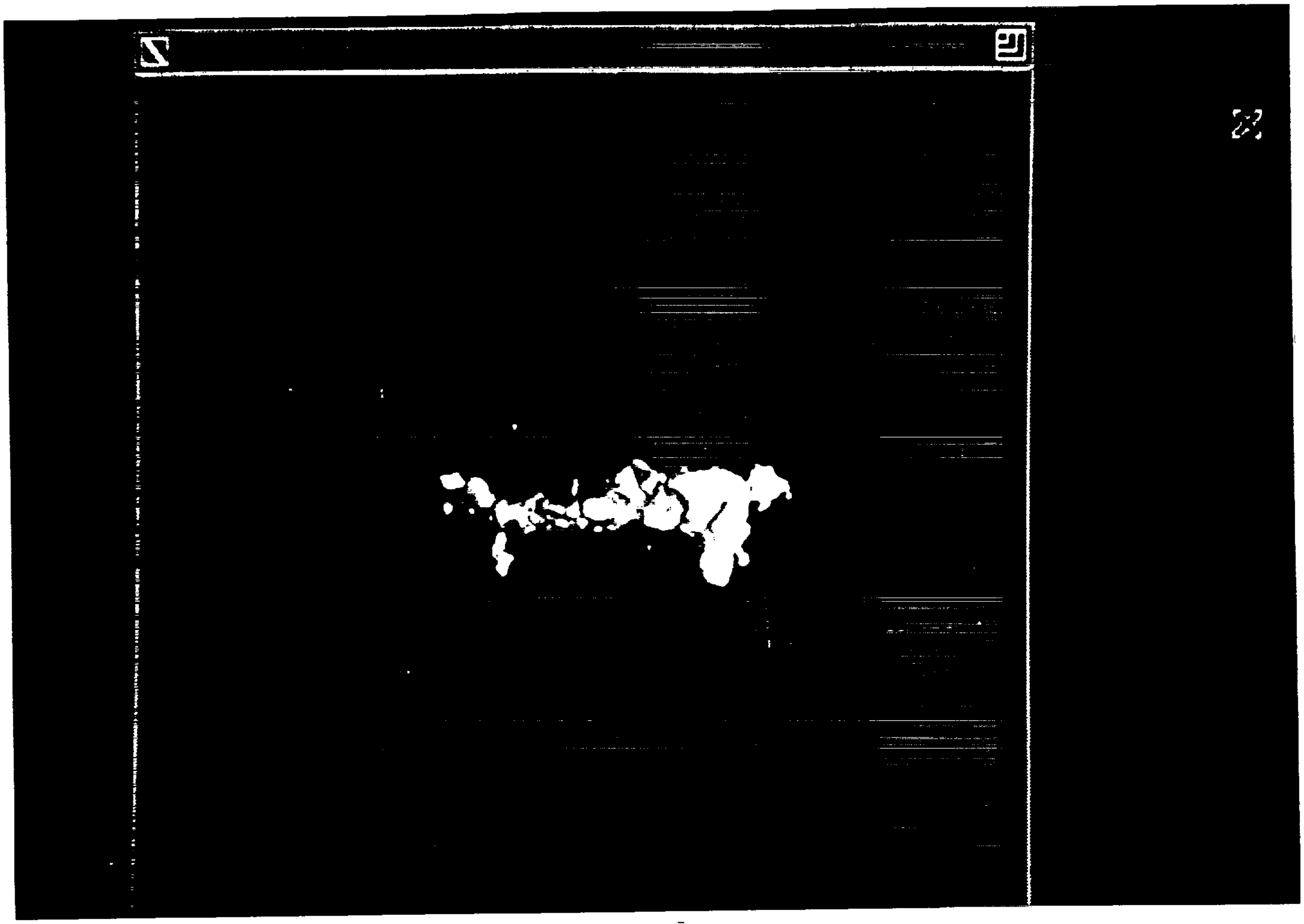


Figure 18: Isosurfaces.

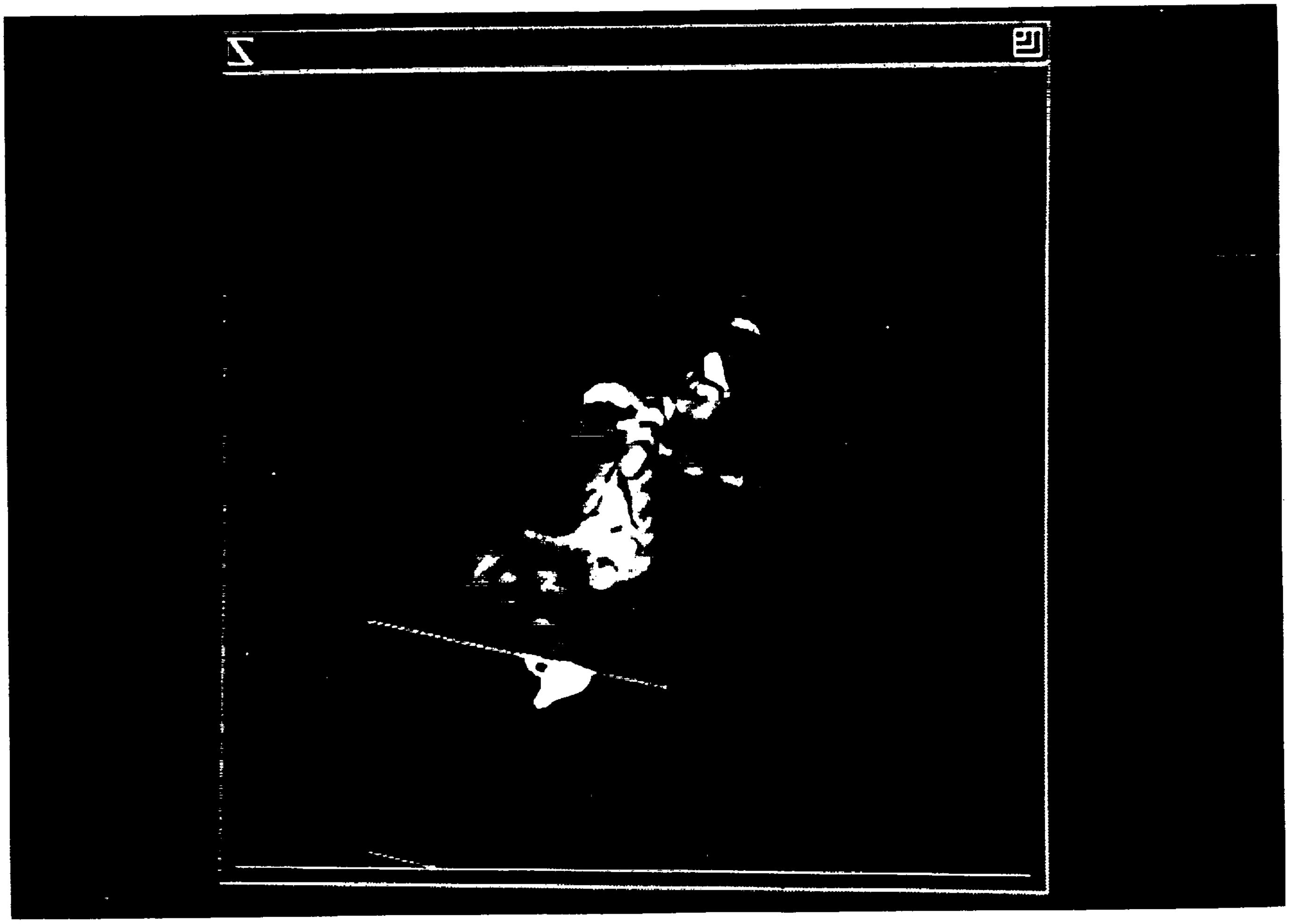
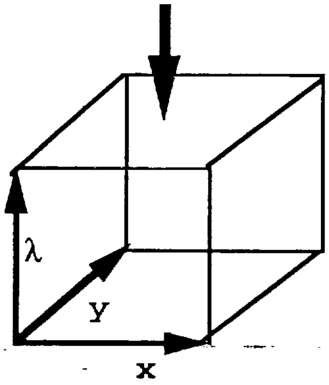


Figure 19: Isosurfaces combined with single image slice

Two transparent isosurfaces
viewing from top



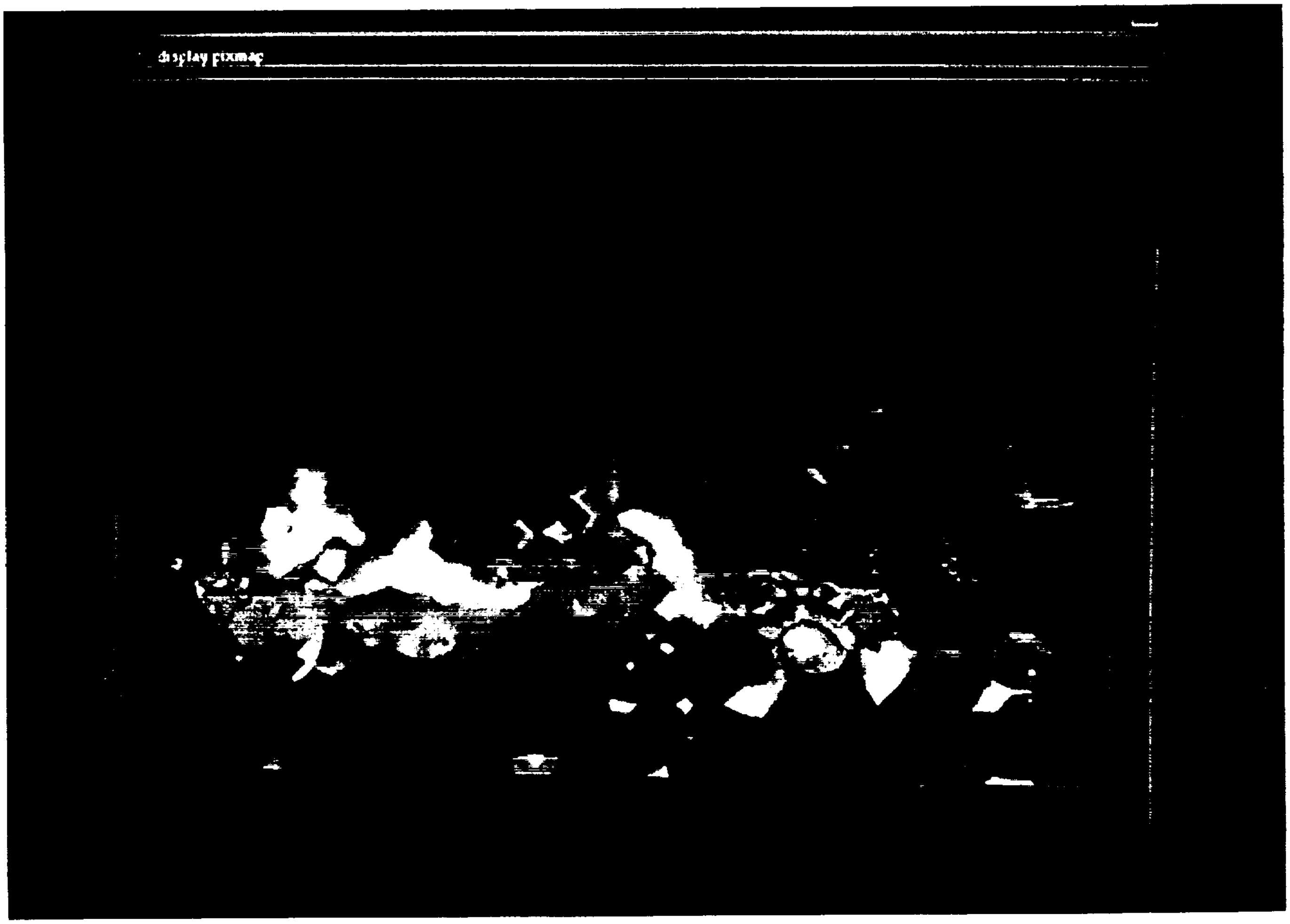
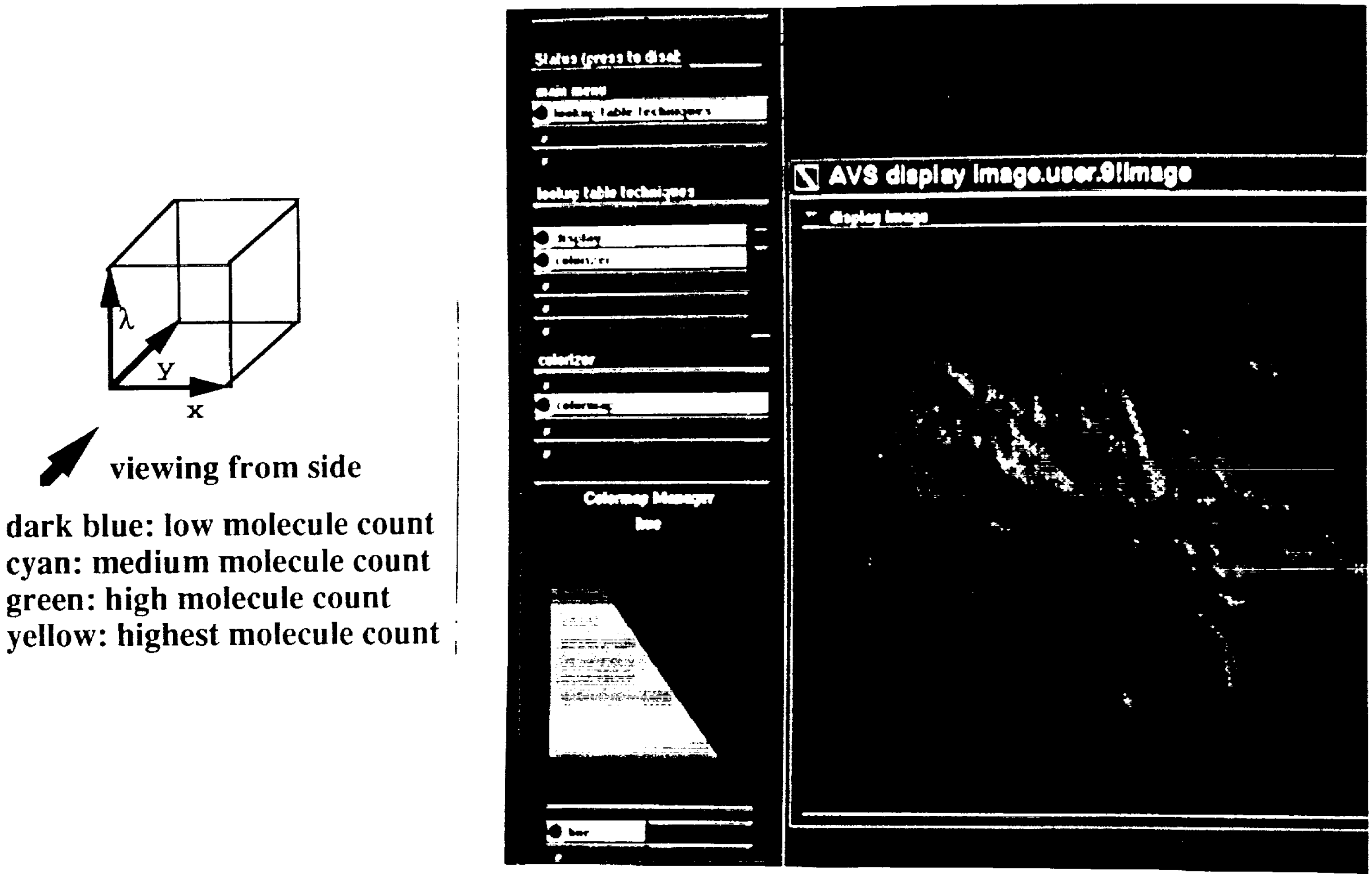
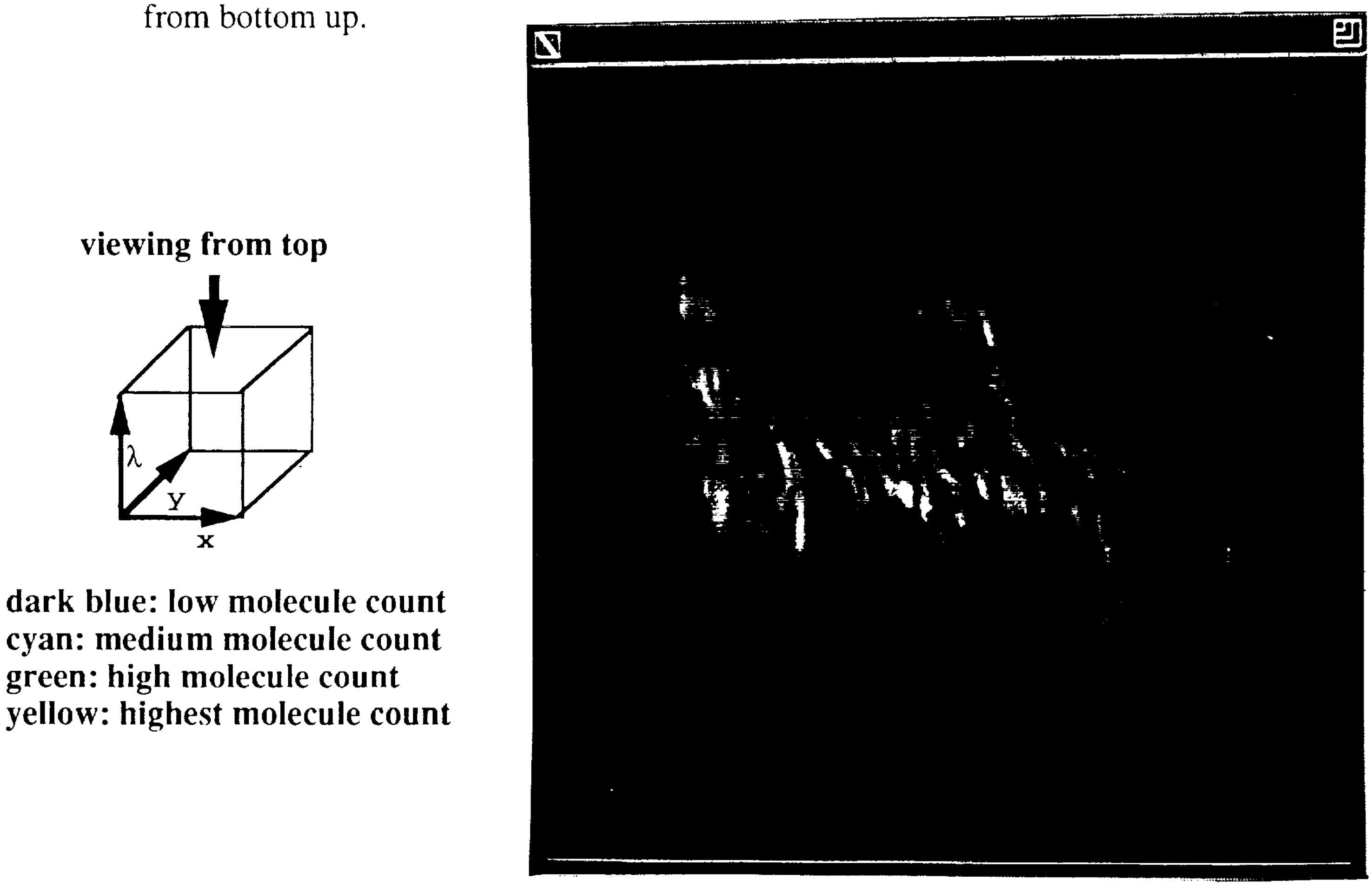


Figure 20: Transparent isosurfaces.



dark blue: low molecule count
cyan: medium molecule count
green: high molecule count
yellow: highest molecule count

Figure 21: Translucent rendering of the cube by looking at the data from one side: one spatial dimension increases to the right, the frequency increases from bottom up.



dark blue: low molecule count
cyan: medium molecule count
green: high molecule count
yellow: highest molecule count

Figure 22: Same data cube as shown in Figure 21 but looking from top down onto the cube

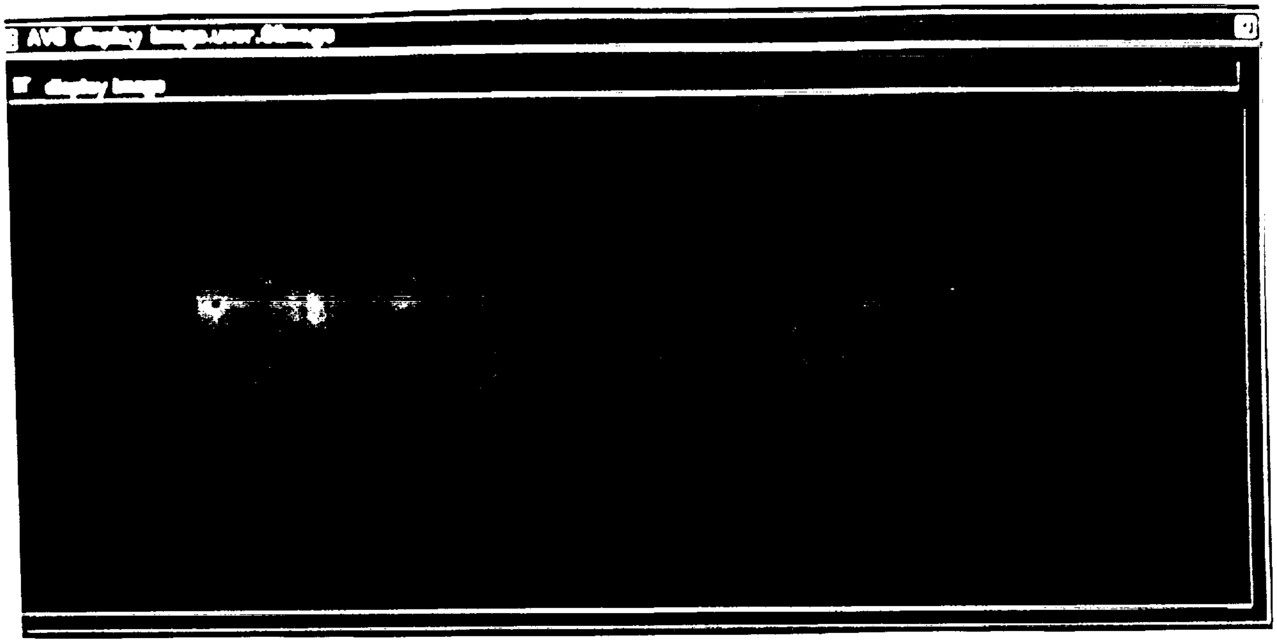


Figure 23: Use of colors relevant to the frequency content: high frequencies are colored blue, low frequencies are colored red. During rotation of the display, the viewer will therefore be constantly aware of the frequency range s/e is looking at.

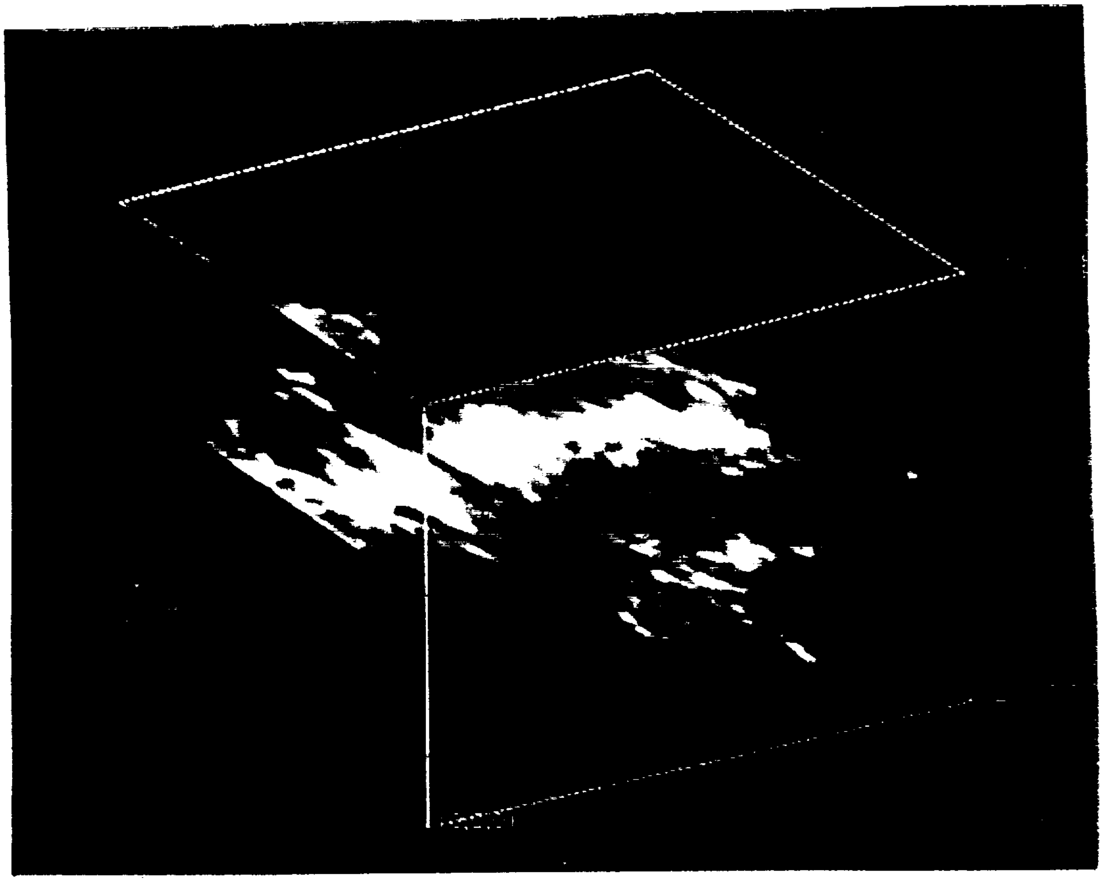


Figure 24: Four slices cutting through the cube parallel to the x/y plane, enhancing the interpretation of the movement of the cloud through frequency.

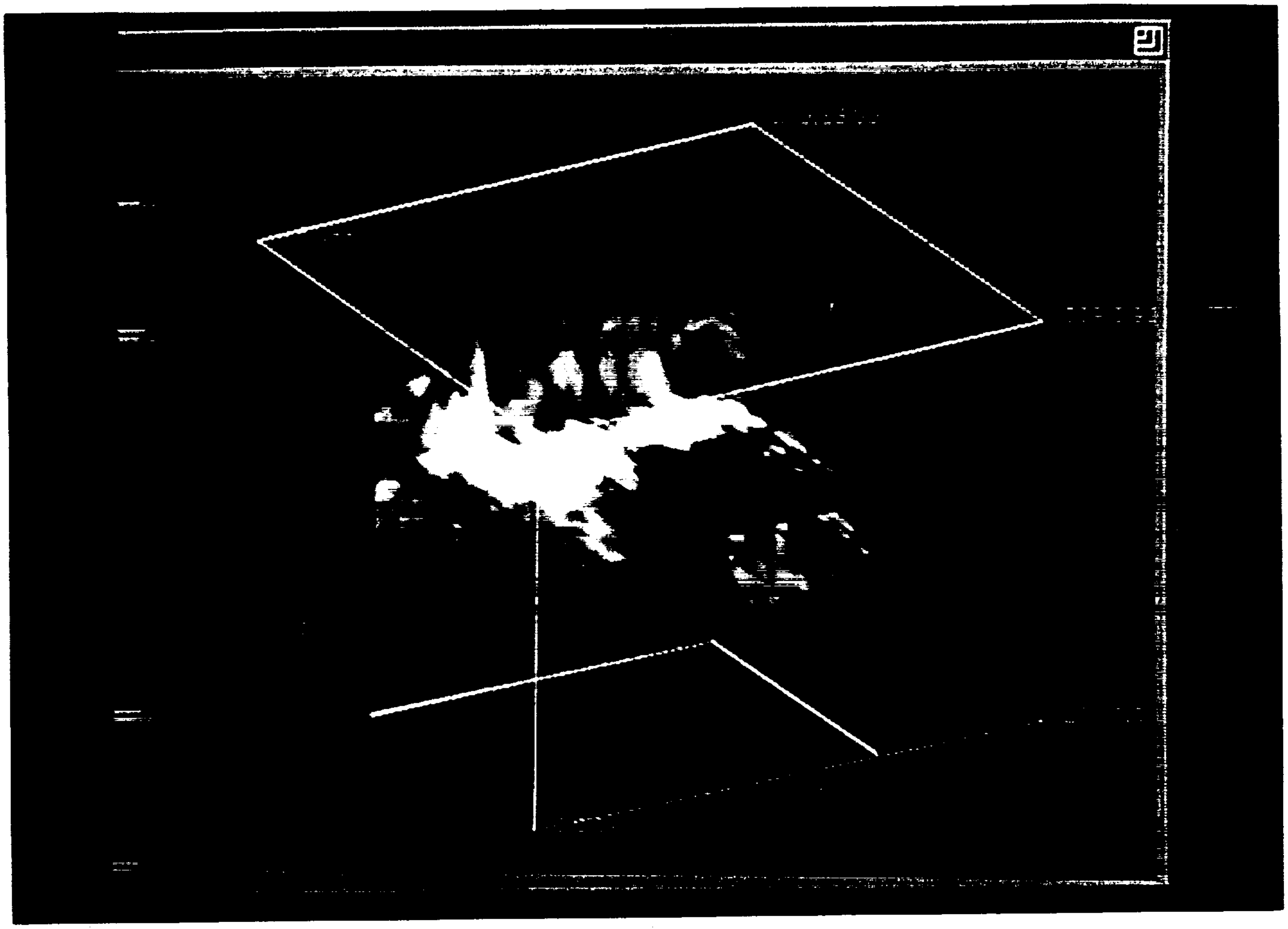


Figure 25: Three orthogonal slices through the data, intersecting in the center of the cube.

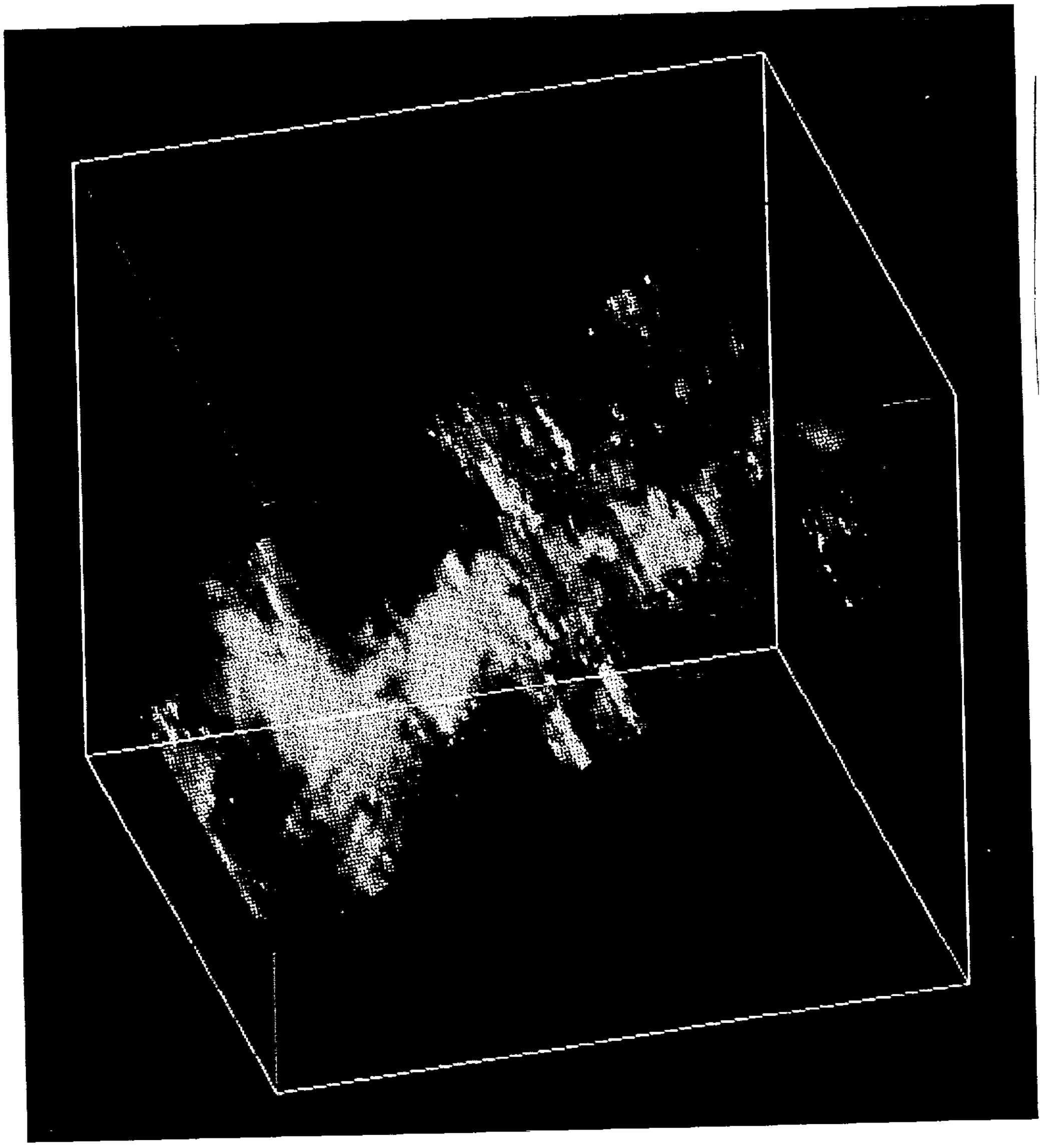


Figure 26: Arbitrary slice through a data cube.

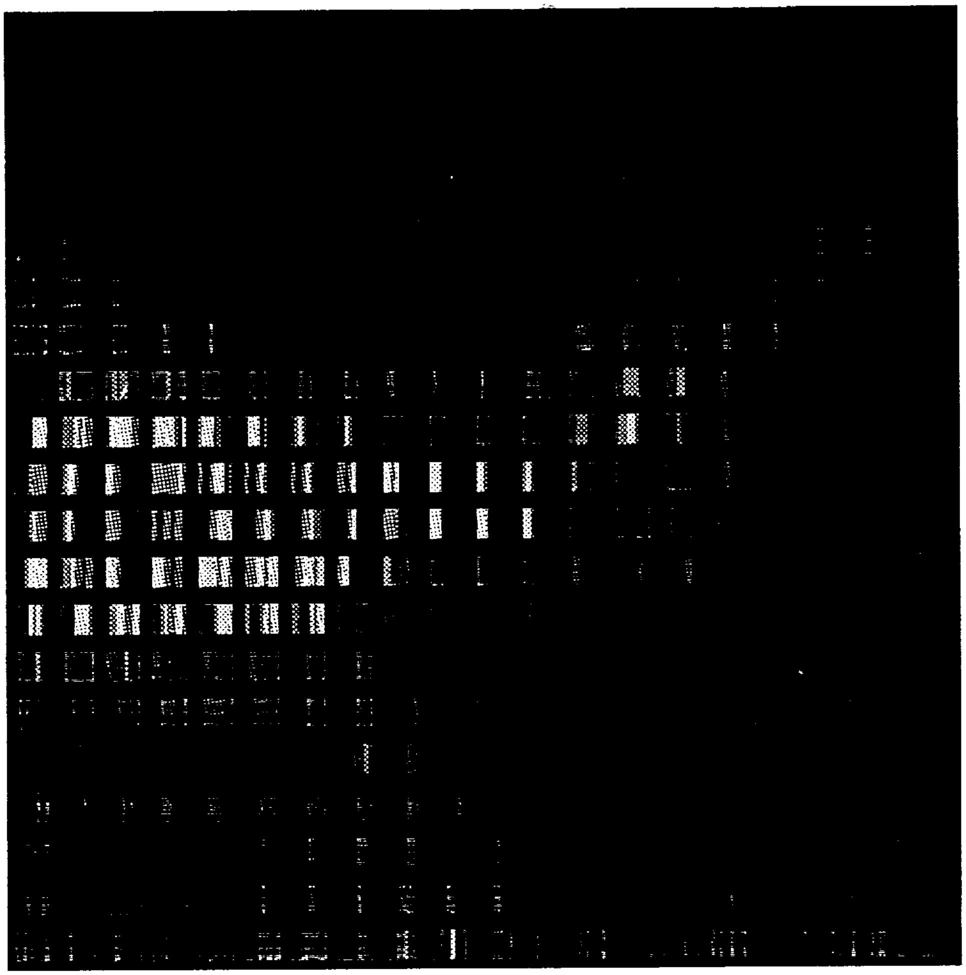


Figure 27: Glyph representation of nine consecutive slices: color slices are used insidė each red square to indicate various spectral responses at each spatial location.

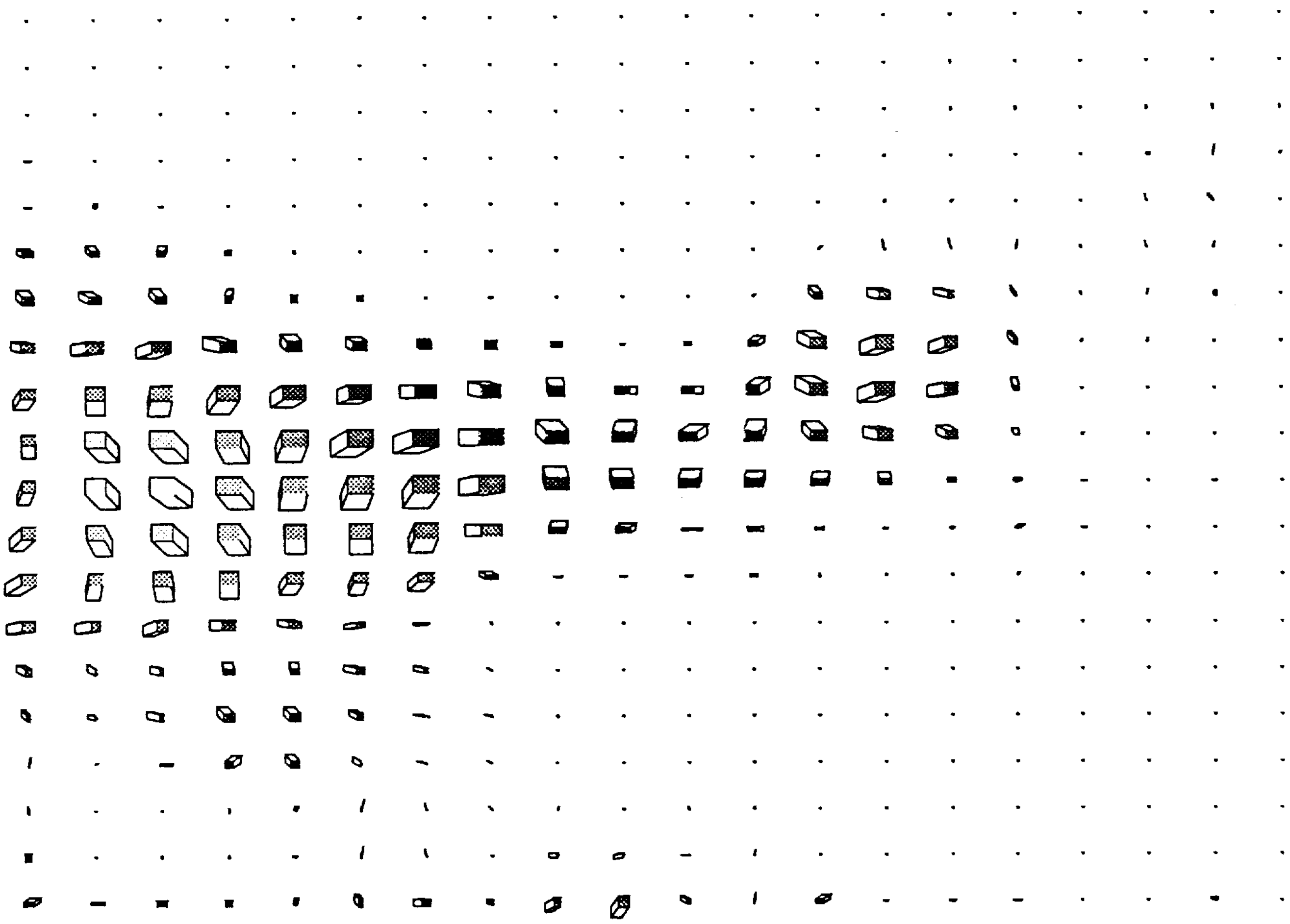
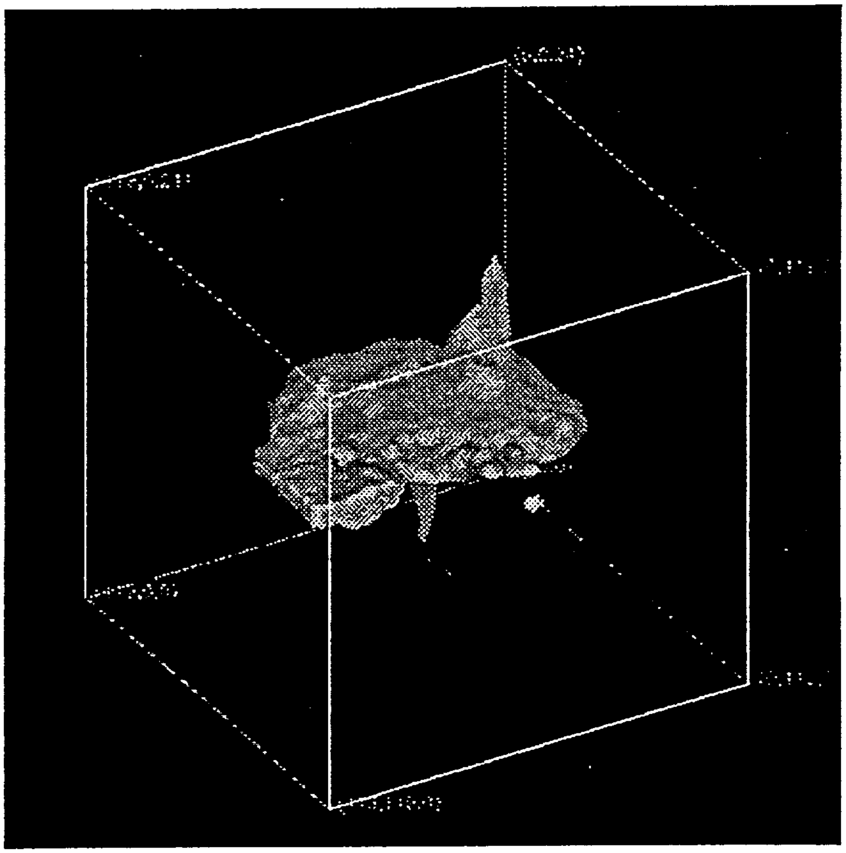


Figure 28: Each five data values are being encoded into five characteristics of a cube: width, height, depth, color and view point.



```
Objects enclosing values higher than 2.00000

Object 1: 45004.1
Object 2: 2.00966
Object 3: 46.4203

Total number of objects: 3
Total sum for all objects: 45052.5
```

Figure 29: Three isosurfaces are displayed and a corresponding count of flux values inside these isosurfaces has been calculated.

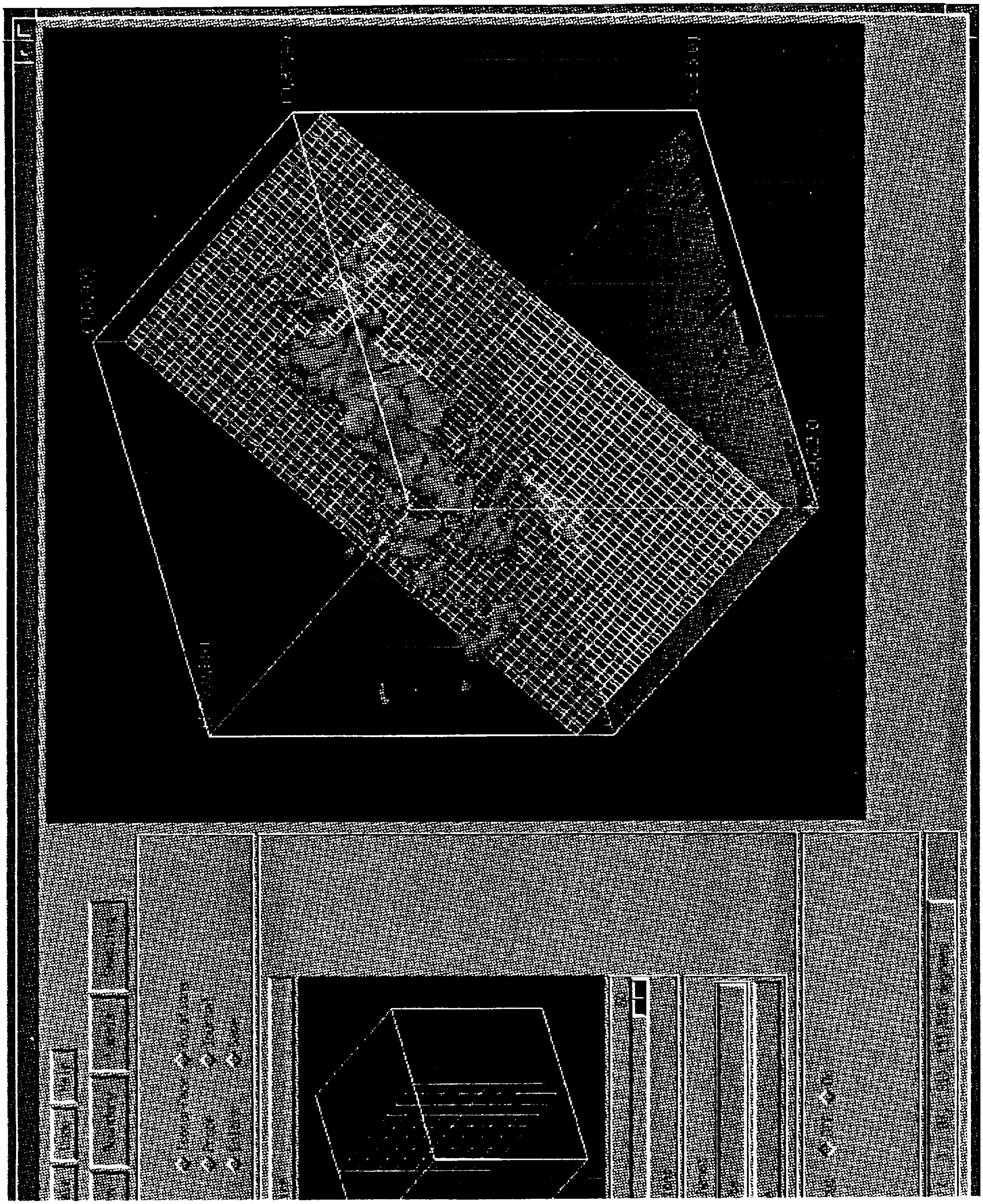


Figure 30: Grid in an arbitrary plane through an isosurface. The grid aids in probing data values inside the data cube in relation to the isosurface.

ORIGINAL PAGE IS
OF POOR QUALITY

4. History of Work Progress (1989 to 1993)

The project started in Fall 1989. At that time the computer environment at the Center for Astrophysics and Space Astronomy (CASA) was mainly based on Microvaxes and VAXstations (Vms operating system), even though two DECstations 3100 under the operating system Ultrix had recently been purchased. IDL with the operating system Vms provided the main software development environment at CASA.

At the start of the project major changes in the hardware and software choices of the future could be predicted: scientific computing environments were (slowly) changing from the Vms environment to RISC/Unix workstations. The Unix workstations provided a desirable price-performance ratio, high-resolution and color on their monitors, affordable peripherals, and much room for improvement in each of these areas. While the astrophysical community in general agreed that RISC/Unix was here for the benefit of scientific computing, "porting software" to new platforms was dreaded. It was the lack of funding for adapting software to a different operating system (and different programming languages) that slowed the switch from Vms to Unix. CASA faced the same problems as other scientific centers of the same size: while the body of software was not as extensive as in larger centers, the necessary programming work on top of ongoing software development was a challenge. Current scientific analysis could not be put on hold for six months to a year while the software was adapted to a new platform. Until June '89, when two DECstations 3100 (Ultrix) were acquired, CASA had strictly operated with DEC equipment and under the Vms operating system. Starting 1989, the policy at CASA was set towards a slow move into the Unix operating domain, in order to take advantage of the superior hardware of RISC architecture, while not interrupting the ongoing scientific work at CASA.

At the start of this project therefore, when all in-house developed and most public domain and commercial software used at CASA still operated only under Vms, the choice was made to acquire a VAXstation 3100/ Model 38 (Vms). Development work was to be done under IDL. The VAXstation joined the existing VAXcluster consisting of two Microvaxes, one VAXserver, 10 VAXstations and two image display stations as node "COSMOS". Cosmos was configured to run Vms and DECwindows (DECs flavor of X-windows). The choice of Vms allowed us to work in an alive and active software environment, even though a future move towards Unix was expected during the project. In participating the later porting to Unix, a platform-independent IDL version was used. Even so, slight differences in the IDL/Vms and IDL/Unix codes were expected, e.g. the use of directory names. Code was therefore carefully developed to use features common to both operating systems; indispensable differences were solved by CASE statements specifying a different solution for each operating system.

The first version of STAR (Scientific Toolkit for Astrophysical Research) was developed, tested and used in the Vms environment between 1990 and 1991. During 1991 a new version of IDL was developed by Research Systems, Inc. to allow a more flexible and extensive use of widgets. Therefore during 1992, STAR was rewritten to take advantage of this new IDL version, improving the appearance and functionality of the user interface, and at the same time making the final switch to Unix. Unfortunately, the local data bases made available under IDL/Vms were not compatible with the Unix environment, so that we lost some of our database functionality with the newest versions of our software. Additionally, and through various unrelated causes, we have lost some of our main users during the last two years (1992/1993). This interrupted the growth of STAR and left STAR in its prototyping phase, giving us a platform to experiment and learn from as well as validate our design goals. STAR in its current form can be ftp'd from the anonymous account at "cetus.colorado.edu" (ip address: 128.138.141.22). Path/filenames to ftp from are "pub/star/star.tar.Z and README.FIRST"[8].

Since 1990 various scientists at CASA and outside the University of Colorado made use of STAR in different ways:

a) Use of STAR: During the course of the project we supported specific needs of scientists by developing new visualization tools and an environment that would benefit individual research demands. As an example, Dr. Edward Brugel, Dr. Robert Stencel, and Dr. John Bally at CASA and their students were strongly supported by STAR's broad functionality (see sections 3 and 6). Outside visitors, like Muriel Taylor from Goddard Space Flight Center, came to CASA to test STAR with their own data.

b) Use of STAR's software: Several scientists and graduate students at CASA (e.g. John Saken) as well as centers outside the University of Colorado (e.g. COBE software development group) have adapted the user interface design or analysis software modules into their own environment. We have allowed this way of using our software as we realized that many scientists design, program and maintain their own code and feel flexible in their own environment. Basing STAR on IDL, a much used software platform in astrophysics, has allowed a widespread use of STAR's modules.

c) Design principles of STAR: More than on software development itself we focused on making general software development guidelines available to the astrophysical community: e.g. how to build user interfaces for astrophysical systems; how to decrease the complexity of large software environments; how to improve

[8] In case of problems please contact Janet Shaw (jes@qso.colorado.edu).

on available visualization modules. The multitude of resulting publications (see section 6) proves our emphasis on this issue.

5. Recommendations to the Sponsors

In order to underline some of the lessons learned during the project, the author would like to emphasize two inherent propositions made in this report:

- the integration of visualization into the data analysis process; and
- the collaboration between astrophysicists and computer scientists.

Integration of visualization into the data analysis process

When we talk about performing data analysis, we actually mean the execution of a series of processing steps. In this report we have divided data analysis into five main components (see section 2): identification and access of existing information; preprocessing; quantitative analysis; visual representation; user interface. Designing and developing software for any of these components should be done in view of the whole process instead of the individual component. The reason for emphasizing the integration of data analysis components is that astrophysicists need to use *all* aspects of data analysis to answer a scientific question. As long as the software development process of each component is separate from the other components, the user of data analysis software is constantly interrupted in their scientific work in order to convert from one data format to another, or to move from one storage medium to another one, or to switch from one user interface to another. Much frustration is spent this way and time used up that could otherwise be used for scientific work.

Because this report was mainly concerned with integrating visualization software into the data analysis process, a smooth integration of visualization functions into the currently existing analysis environment was stressed. However, visualization of scientific data is meaningless on its own: data has to be documented, calibrated and measured together with their visualization in order to reveal their meaning in respect to a scientific question.

Collaboration between astrophysics and computer sciences

Closer collaboration of astrophysicists and computer scientists will increase productivity and quality in astrophysical research. The competitiveness of research, current funding structures, and historical separation is hampering the collaboration between these groups. A certain indifference of each others needs and contributions leads to commercially available visualization systems that do not cover the needs of astrophysicists (or other scientists); an ignorance of new methods available for multi-dimensional visualizations; or misleading assumptions in introducing new technologies.

Recommendations to support necessary changes include appropriate actions in the research funding structure, the sponsorship of computer scientists to spend time in the environment of astrophysicists, or the sponsorship of computer scientists to develop and teach tutorials on new methods and technologies at NASA's facilities.

Closer collaboration is to the benefit of both parties, and efforts such as the ones instigated through the Center for Excellence of Space Data Information Sciences (CESDIS) are expected to make a difference in the future.

6. Dissemination of Project Activities Outside the University of Colorado

6.1 Publications in journals or books

Domik, G., Brugel, E.W., Stencel, R.E., Vasudevan, S., Pang, J., 1990, *Workstation based Preprocessing of IRAS Skyflux Images,* Publications of the Astronomical Society of the Pacific, October 1990.

Domik, G. O. and Mickus-Miceli, K.D. 1992, *Design and Development of a Data Visualization System in a Workstation Environment,* J. Microcomputer Applications, Vol. 15, pp. 81-88.

Domik, G.O. and Mickus-Miceli, K.D., 1992, *Software Design and Development in a Scientific Environment: Lessons Learned During the Development of STAR, an Astrophysical Analysis and Visualization Package,* Astronomical Society of the Pacific Conference Series, Volume 25, Astronomical Data Analysis Software and Systems I, ed. by D.M. Worrall, Ch. Biemesderfer, and J. Barnes; pp. 95-99.

E.W. Brugel and R. Fesen, *Optical Imaging and Spectroscopy of Three New Northern Hemisphere Herbig-Haro Complexes,* in preparation.

6.2 Publications in proceedings

Domik, G., Vasudevan, S., Pang, J., 1990, Brugel, E.W., Stencel, R.E., *Applications of IRAS Preprocessing at the Workstation,* 176th meeting of the American Astronomical Society, BAAS Vol. 22, No. 2, 1990.

Mickus, K., Domik, G., Brugel, E., Ayres, T. 1990, *STAR-- A Scientific Toolkit for*

Astrophysical Research, 176th meeting of the American Astronomical Society, BAAS Vol. 22, No. 2, 1990.

K. Mickus, E. Brugel, G. Domik, T. Ayres, 1991, *A Case Study: Multi-Sensor Data Analysis of HH Objects via STAR*, BAAS Vol. 22, No. 4, 1991.

G. Domik and K. D. Mickus, 1991, *Visualization in the Analysis Cycle of Observational Data*, Proceedings of the Gesellschaft für Klassifikation, Salzburg, Feb. 25-27, 1991.

E.W. Brugel and R. Fesen, 1991, *Discovery of Three New Herbig-Haro Objects*, BAAS, Vol. 23, No. 2.

E. Overgard, R. Stencel and K. Mickus, 1991, *Workstation Based Analysis of IRAS Views of OB Star Associations*, 103rd ASP meeting, Laramy, Wy..

Domik, G., 1992, *Designing User-Centered Interfaces for Astrophysical Software*, Workshop Proceedings on "User Interfaces for Astrophysical Software", April 14-15, 1992, Goddard Space Flight Center, Greenbelt, MD, sponsored by NASA Headquarters, Astrophysics Division Workshop.

Domik, G. , 1992, *Visualization of Multi-Dimensional Arrays in Astronomy*, Conference on "Astronomy from Large Databases II", Haguenau, France, September 14-16, 1992.

6.3 Reports

Ruiz, J. and Domik, G., 1990, *Use of Color Coding Techniques to Search for Herbig-Haro Objects in the Infrared*, internal REU (Research Experience for Undergraduate Program of the National Science Foundation) report.

K. D. Mickus, 1991, *Participatory User Interface Design for Scientific Visualization Systems*, Masters Thesis, University of Colorado, Department of Computer Science, Boulder, Colorado, 80309-0430.

Domik, G., 1992, *A Case Study in Astrophysical Data Visualization*, Technical report CU-CS-622-92, University of Colorado, Department of Computer Science, Boulder, CO. 80309-0430.

Kahn, Bredekamp, Cheeseman, Domik, Knighton, Treinish, and Weir, 1992, *How Best to Create Interactive Data Analysis Tools That Allow the User to Go from the Pictures to the Numbers*. Special Report to the Communications and Information Systems Division, NASA, March, 1992.

6.4 Participation at workshops relating to project work

User Interfaces for Astrophysical Software, a workshop sponsored by the Astrophysics division of NASA, April 14-15, 1992:

Going from the pictures to the numbers, a workshop sponsored by the Communications and Information Division Systems of NASA, February 1992.

6.5 Talks

A series of about 20 talks were given over the past four years to report on project work and to solicit feedback from scientists about the goals and results of the project. About an equal share of presentations were given in front of scientists (e.g. at the Jet Propulsion Laboratory; at NASA Goddard Space Flight Center; at the Department of Meteorology at the University of Innsbruck, Austria) and computer scientists/engineers (e.g. Department of Informatics, University of Linz, Austria; Storage Tech, Louisville, Colorado; Department of Computer Science, University of Illinois, Urbana-Champaign and National Center for Supercomputing Applications (NCSA)). Consequently a better understanding of the issues surrounding the software environment of data analysis, and specifically visualization software, emerged.

References

Ayres, T., Brugel, E., Domik, G., 1989, *Visualization Techniques to Aid in the Analysis of Multi-Spectral Astrophysical Data Sets,* Proposal to NASA, Astrophysics Division.

Mackinlay, J., 1986, *Automating the Design of Graphical Presentations of Relational Information*, ACM Trans. Graphics, Vol. 5, No. 2, April 1986, pp. 110-141.

McKim, R.H., 1980, *Experiences in Visual Thinking*, PWS Publishers, second edition.

K. D. Mickus,1991, *Participatory User Interface Design for Scientific Visualization Systems*, Masters Thesis, University of Colorado, Department of Computer Science, May 1991.

Robertson, P.K., 1990, *A Methodology for Scientific Data Visualization: Choosing Representations Based on a Natural Scene Paradigm,* Proceedings of the Conference on Visualization '90, October 23-26, San Francisco, pp. 114-123.

Wehrend, S., and Lewis, C., 1990, *A Problem-Oriented Classification of Visualization Techniques*, Proceedings of the Conference on Visualization '90, October 23-26, San Francisco, pp. 139-143.

Appendix A:

Workstation-based Preprocessing of IRAS Sky-Flux Images

PUBLICATIONS OF THE ASTRONOMICAL SOCIETY OF THE PACIFIC
102: 1167–1180, October 1990

WORKSTATION-BASED PREPROCESSING OF IRAS SKY-FLUX IMAGES

GITTA DOMIK, EDWARD W. BRUGEL, AND ROBERT E. STENCEL

Center for Astrophysics and Space Astronomy, University of Colorado
Campus Box 391, Boulder, Colorado 80309-0391

JINZHONG PANG

Department of Physics, University of Colorado, Campus Box 390, Boulder, Colorado 80309-0390

AND

SUBRAMANIAN VASUDEVAN

Electrical and Computer Engineering Department, University of Colorado
Campus Box 425, Boulder, Colorado 80309-0425

Received 1990 March 28, revised 1990 June 28

ABSTRACT

We have developed and implemented computer algorithms to remove two types of degradations in IRAS sky-flux images: slowly varying background illumination (strongly effected by the presence of zodiacal light) and periodic stripes. This paper discusses both algorithms in detail and shows results of its use on various sky-flux images.

Focus of the work was on the implementation within a workstation environment and its value as a preprocessing tool for researchers: Speed of the process, usability of the programs, and correctness of the results were our main goals in developing these tools.

Key words: IRAS sky-flux images–preprocessing–destriping–flattening–Fourier space algorithms

1. Introduction

Researchers working with currently available IRAS (Infrared Astronomical Satellite) sky-flux data have been hampered in their work due to the undesirable effects of slowly varying background illumination within sky-flux plates and of periodic distortions, producing sinusoidal stripes in the image.

The Infrared Processing and Analysis Center, IPAC, is trying to effectively remove these degradations by reprocessing original source data. New sky-flux data are expected in the near future; however, we believe that an independent assessment of background-removal techniques may offer researchers more understanding of the methods involved.

Because products from the IRAS provide major research data sets, it is necessary to remove degradations to utilize sky-flux images to their fullest extent. We succeeded in developing a procedure that would remove these degradations but still preserve essential image information. The procedure we implemented is fast, fully automated, and programmed in Interactive Data Language (IDL) which is becoming a widely used software platform in the scientific community (Stern 1984; Varosi, Landsman, and Pfarr 1990).

Section 2 gives a mathematical and technical explanation of the algorithm for flattening the uneven background flux; Section 3 explains the algorithm to remove stripes and compares the method to the one developed by Van Buren (1987); Section 4 summarizes the effect of the restoration techniques through several examples. The thus-restored IRAS sky-flux images are used at the Center for Astrophysics and Space Astronomy (CASA) as a basis for various astrophysical research efforts, such as in multiwavelength analysis (Saken, Shull, and Fesen 1990), and an investigation into the use of IRAS images for the detection of Herbig-Haro objects (Domik and Brugel 1990).

2. Flattening of Background Flux

Variation of the position angle between the data-collecting platform and the Sun is the main cause of gradient background flux increase or decrease. This causes problems in the visual and numerical analysis of sky-flux images, because pointlike and extended sources of similar power cannot be compared anymore numerically or visually if they appear sufficiently apart in the image. A sky-flux image, I, degraded by varying background flux, can be approximated by equation (1):

$$I(x,y) = F(x,y) + Z(x,y) \ , \qquad (1)$$

where F is the flux image without degradations, Z denotes the slowly varying background values, and (x,y) indicate the two spatial dimensions.

Figure 1(a) shows an example of sky-flux plate 75 (25 μ), centered at a right ascension of 4 hours and a declination of 15°. Flux values at the left upper corner are elevated by an average amount of $6.1\ e + 06$ Jy/sr compared to the lower-right corner.

The amount of relative flux elevation over the image can be approximated by measuring average flux values in the four quadrants of the sky-flux image, after point sources have been removed. Removal of point sources is performed by a median filter with a large (e.g., 11 × 11) window size. The degradation is then modeled from four points of estimated background flux as either a plane (least-squares fit) or a hyperbolic paraboloid (Fig. 1(b)). Approximations of higher order might remove structures not part of the background. Removal of the degradation is accomplished by simply subtracting the background model from the image. In order to keep flux values in their intended range, we elevate the result to the average of the original image.

The algorithm can be summarized by the following pseudocode:

```
1. $I' = F[I]$, with $I$ = orig image (degraded), $F$ .. median filter
2. $z_i = \mathrm{AVG}[Q_i[I']]$, with $Q_i[I'] = i$th quadrant of $I'$, AVG ... average function
3. $Z = \mathrm{BACKGROUND\ MODEL}[z_i]$, for $i = 1,4$
4. $F \quad I \quad Z - \mathrm{AVG}[I]$
```

Implementation of the above algorithm in IDL is fully automated and fast. Availability of information about minimum and maximum flux and size of image as provided with images previously distributed by IPAC is assumed. The result of flattening the image in Figure 1(a) is presented in Figure 1(c). Computation times are summarized in Section 4.

3. Removal of Sinusoidal Stripes

The cause of these degradations lies in the scanning nature of the survey, nonuniform response across each detector, and differences between detectors of the IRAS scanner (Beichman *et al.* 1985; Kennealy *et al.* 1987; Whaley 1989). The differing response from each individual detector gives rise to a high-frequency variation and is strongest in bands 1 and 2. Each separate pass of the sensor across the image caused stripes of lower frequencies (approximately 9–11 stripes per image) and is again most noticeable in bands 1 and 2. Removal of periodic stripes is important for the recognition of faint stars and useful background flux information that might otherwise not be detectable in the image.

Periodic features that show up in the form of sinusoidal waves can be more easily identified in the frequency domain than in the spatial domain, because sinusoidal features transform to impulses in the frequency domain. An impulse can be easily identified and removed without accidentally eliminating important signals as well.

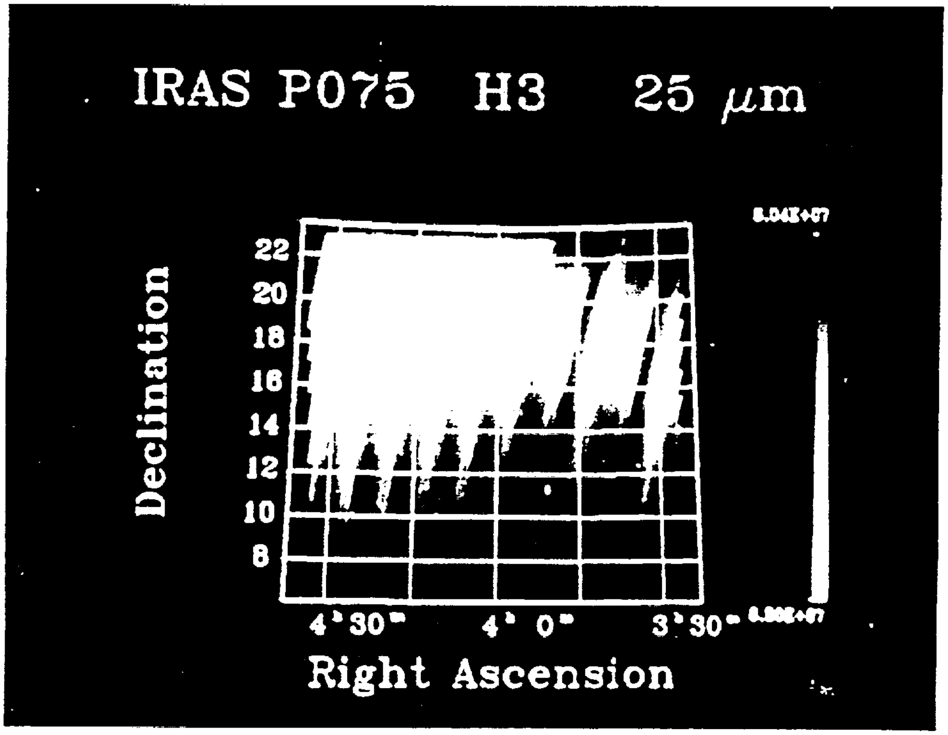


FIG. 1–Sky-flux plate 75, HCON 3, band 2. (a) Shows the sky-flux plate as currently distributed by IPAC. Average flux values at the upper-left corner are elevated by an average of $6.1\ e + 06$ Jy/sr compared to the lower-right corner.

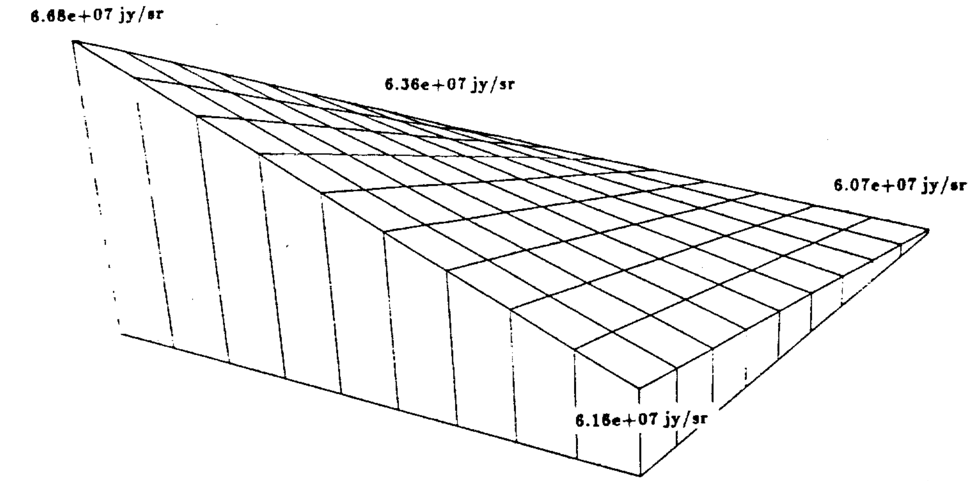


FIG. 1–Sky-flux plate 75, HCON 3, band 2. (b) Shows the background modeled as the surface of a hyperbolic paraboloid.

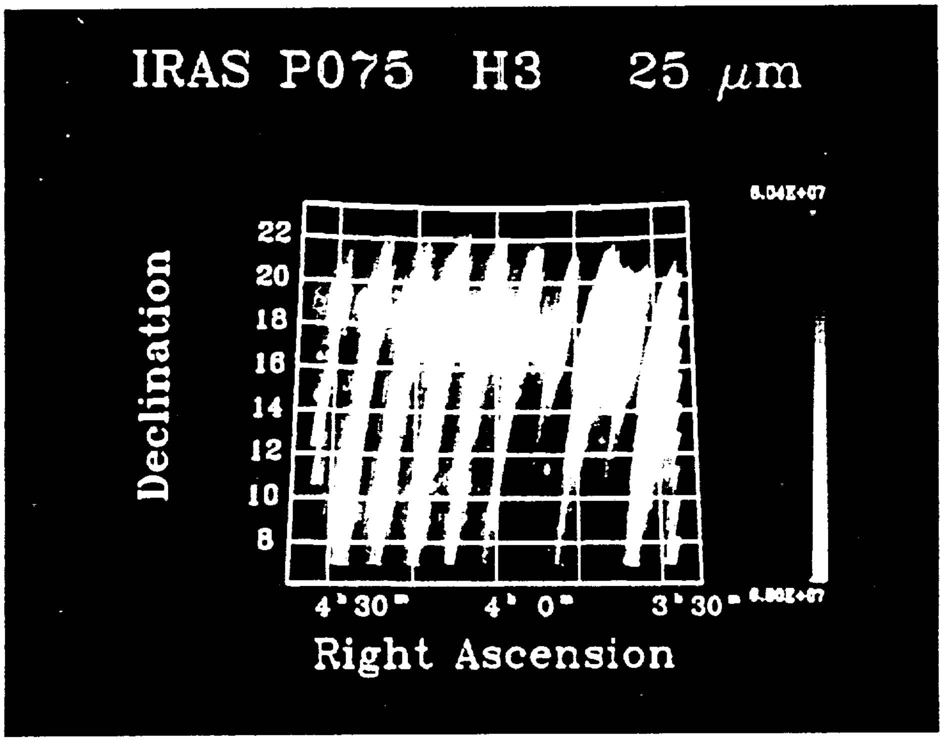


FIG. 1–Sky-flux plate 75, HCON 3, band 1. (c) Shows the sky-flux plate after removing background surface.

Unfortunately, the Fourier components of point sources spread over most of the frequency domain. Table 1 summarizes the information content of an IRAS sky-flux image and its presentation in the spatial and frequency domain.

In order to remove periodic features but none of the image signals requires that a precise elimination process be applied. We have carefully reviewed the algorithm by Van Buren (1987) and found it very useful in several ways:

1. By removing point sources (using upper and lower cutoff values) before the filtering process one reduces the danger of accidentally removing signals from the point sources during the filtering process.

2. By removing only small "wedges" in the frequency domain one again reduces the risk of removing too much of the signal.

We have improved on the above algorithm in two major ways:

1. In order to remove point sources from the original image the user does not need to prespecify cutoff values; instead, the algorithm is able to identify pointlike objects and remove them from the image. User-defined specification of cutoff values caused two problems: for one, the user needed a clear understanding of what range of flux

TABLE 1

Information Content of IRAS Sky-flux Images

Image Feature	Spatial Domain	Frequency Domain
Stars	Pointlike features (sharp, narrow Gaussians)	Spread all over domain (wide Gaussians)
Stripes	Periodic features	Impulses in high and low frequencies
Nebulae, other background features	Various forms	Mainly in lower frequencies

values represented periodic striping and, therefore, needed to numerically review a number of image profiles; second, even with specifying a perfect upper cutoff value faint point sources were often hidden in the lower values of the periods of the stripes.

2. The program is able to identify a range of angles signifying the slopes of the stripes automatically. This proves very helpful in the overall restoration algorithm in two ways: first, because the program is able to identify the slopes of stripes more accurately and therefore leads to less signal removal later on; second, this step allows the algorithm to become fully automated; there is no need for any type of measurements before the restoration algorithm is started.

The destriping algorithm presents itself in six stages:

Step	Pseudocode	Explanation
1.	$B = F - P$; F ... flux image	remove point sources from image
	P ... point sources	
2.	$\alpha = \Phi(B)$	calculate direction of slopes for stripes
3.	$\mathbf{B} = \mathcal{F}\{B\}$	transform background into frequency domain
4.	$\mathbf{B}' = \mathbf{B} \cdot \mathbf{W}(\alpha)$	apply appropriate filter over areas affected
	$\mathbf{W}(\alpha)$ ... frequency filter	
5.	$B' = \mathcal{F}^{-1}\{\mathbf{B}'\}$	transform background
6.	$F' = B' + P$	add point sources onto background

Details of each step follow.

3.1 *Remove Point Sources from Image*

To explain our procedure in more detail we used sky-flux plate 75 (HCON 3, band 2) as an example. Figure 2(a) shows a profile of the image in its original version, but after flattening the background. The profile is taken at a declination of 9°59′. To separate point sources from the background we first median filter the image with a large window size. The window size of the median filter should be large enough to incorporate the image size of all point sources, but a single "correct" value is not intrinsic to the success of the whole algorithm. The upper cutoff value for point sources is then defined at a fraction of a standard deviation from the median values. Figure 2(b) shows the identification of the background: The broken line signifies values at a distance of $\sigma/8$ from the median value, where σ is the standard deviation in the image; the solid line shows the thus-identified "background image".

The cutoff distance should be large enough to still identify high-frequency degradations as "background" but low enough to identify point sources as such. By default, the system uses $\sigma/8$; however, $\sigma/8$ to $\sigma/4$ have been empirically defined as good values. Because of our dynamic definition of cutoff values, point sources can be identified even inside the periodic stripes. A profile of the features identified as point sources is presented in Figure 2(c).

3.2 *Calculate Direction of Slopes for Stripes*

The automated calculation of slopes defining the stripes is performed using a correlation technique between several image profiles. The image profiles are taken from the median-filtered background image (see Fig. 2(b)); thus, the periodic waves caused by the stripes are easily observable. Correlation is performed by shifting one profile line against another and measuring the difference. The difference translates into an error function, whose minimum value relates to the shift of stripes between the two lines.

Because the slope of the stripes changes in the overall image (specifically at the upper and lower edges of the image compared to its center), a range of angles seems more representative than one specific angle. For example, sky-flux plate 75, HCON 3, band 2 (Fig. 1(a)) was found to have stripes between angles of 7° and 13°.

3.3 *Apply Appropriate Filter over Areas Affected*

After transforming the image to the frequency domain the disturbing impulses are removed. The location of these impulses is found rotated 90 degrees from the slopes calculated for the image stripes (Gonzalez and Wintz 1987). Because we assume a range of slopes, we remove (or suppress) the values within two wedges, as suggested in Van Buren (1987). See Figure 3 for a step-by-step presentation of the filtering technique.

Because we start off with a flattened background, the most prominent features both visually and numerically are the periodic stripes (Fig. 3(a)). They are even more obvious in the frequency domain (Fig. 3(b)), where they give rise to a series of impulses orthogonal to their direction in the spatial domain. After affected areas have been blocked off (Fig. 3(c)), and an inverse Fourier transform is performed, the image shows reduced periodic degradations. However, depending on processes applied to the raw image, stripes hidden in the background before the process may become visible now. Subsequent passes will remove such stripes, but it is up to the scientist to balance

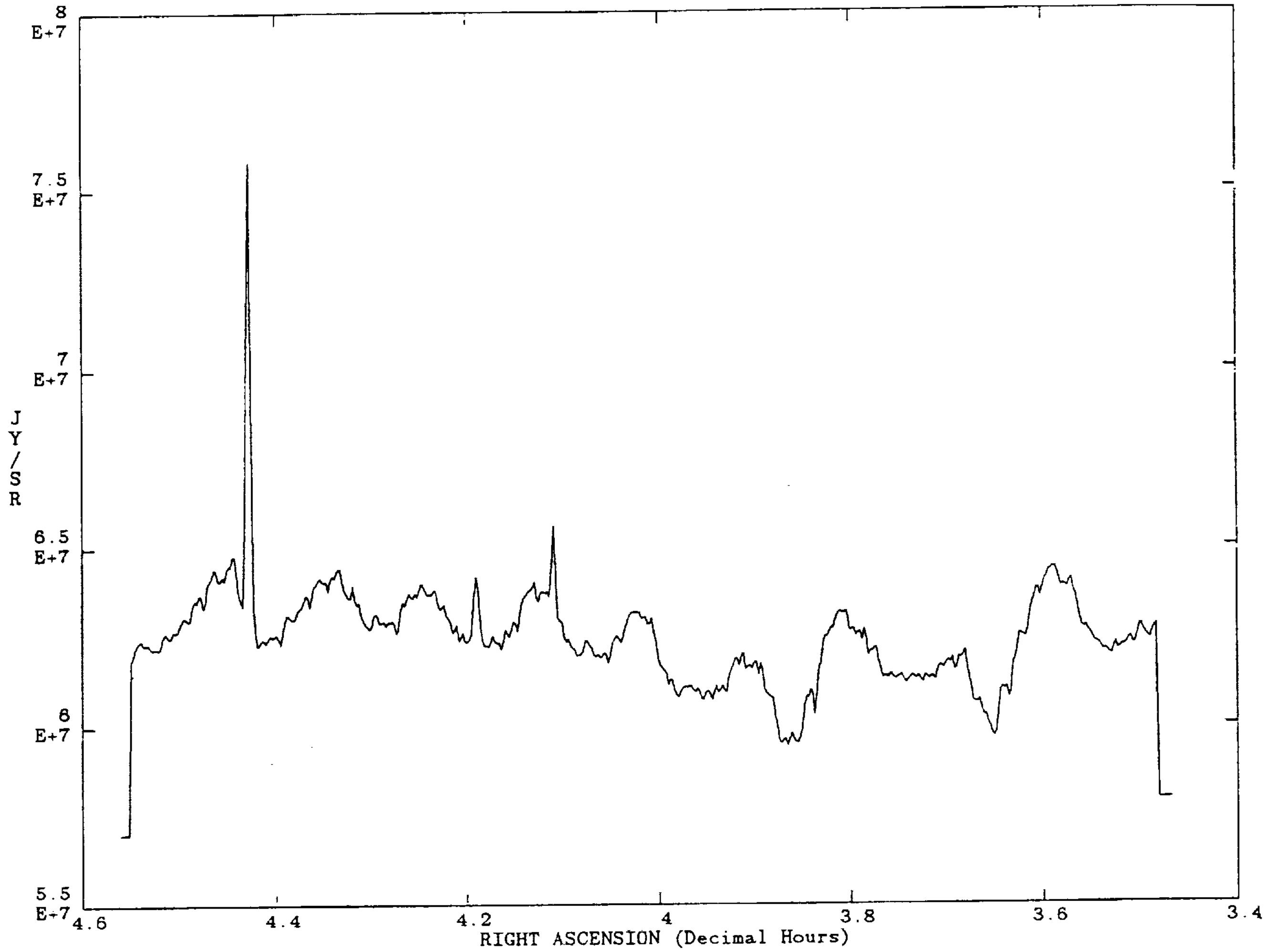


FIG. 2–Profile at a declination of 9°59′ of sky-flux plate 75, HCON 3, band 2, after flattening background. (a) Before destriping.

eventual loss of signal versus the level of destriping desired. Figure 3(d) shows the level of destriping (7° to 18°) chosen by the authors.

Because the original dynamic range in sky-flux images is usually found between 1.0 *e* 07 Jy/sr and 1.0 *e* Jy/sr, this would give rise to high values in the frequency domain. Therefore, we normalize the image values by shifting them into the range of [0..1] before taking the Fourier transform. The Fourier-transformed values then lie between a controlled area instead of being data dependent. This facilitates the filter design, which is either a simple mask to set the values inside the wedges to zero or a specifically designed filter affecting only the disturbing impulses in the frequency domain.

3.4 *Add Point Sources Back onto Background*

After the filtering step discussed above, the background is unfolded into its original dynamic range and point sources are added back into the restored background image. Figure 4(a) shows a profile of the restored image, but before adding point sources. Figures 4(b) and 3(d) show the restored image after adding point sources: Even though the periodic degradations are eliminated, the information of the sky-flux images seems fully contained.

4. Examples of IRAS Preprocessing and Outlook

In order to prove the value of the above-described procedures, we have preprocessed a set of well-published IRAS sky-flux images. Figure 5 shows a series of images from sky-flux plate 77, before and after preprocessing. Note, again, that all the preprocessing is done without user interaction. Overriding the range of slope values by user-identified values before filtering is possible. CPU times for removal of the background is 0.6 minute (Fig. 1(a) to Fig. 1(c)); for destriping we measured 2.6 minutes per 500 × 500 sized sky-flux image (Fig. 3(a) to Fig. 3(d)). CPU times were measured on a heavily used VAXserver 3500.

We believe that the automation of the restoration process, its speed, and its accuracy lend itself to a wide use of applications for researchers dealing with IRAS sky-flux

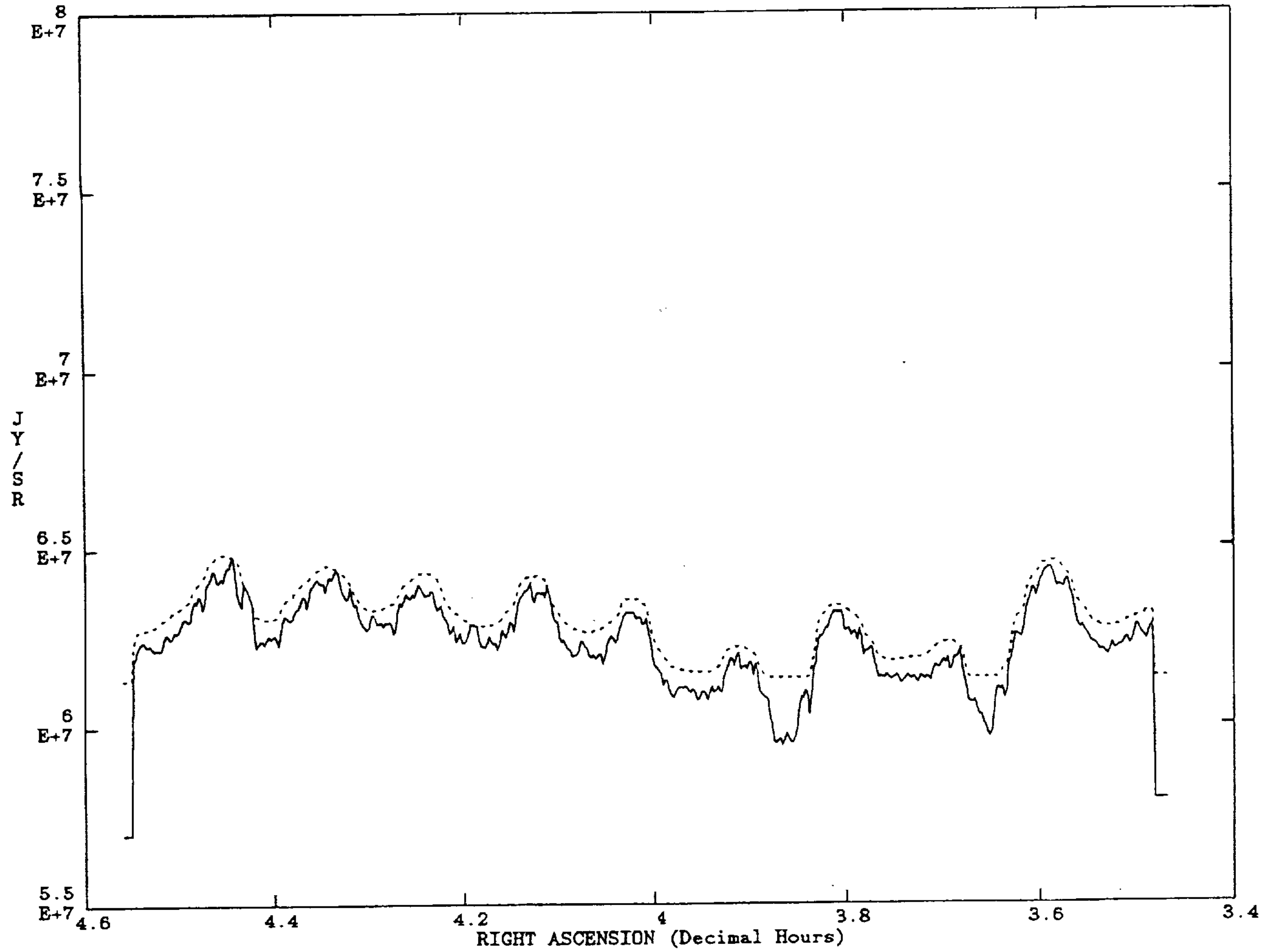


FIG. 2–Profile at a declination of 9°59′ of sky-flux plate 75, HCON 3, band 2, after flattening background. (b) Upper cutoff values appear in dotted lines ((σ/8) removed from median values). Solid line shows "background image".

images. One of the applications of the above-mentioned preprocessing technique at CASA is the use of multiwavelength data in the infrared range to identify Herbig-Haro objects. This research interest evolves from work by E. W. Brugel in the ultraviolet range and will investigate the possibility of color-coding techniques to identify Herbig-Haro objects in IRAS images. Preprocessing as explained in this report is a necessary step if visual interpretation of flux images is to be meaningful. A forthcoming publication will discuss this project in more detail (Domik and Brugel 1990).

Algorithms described in this paper will be made available to the astronomical community as documented and tested IDL programs through the "IDL Astronomical User's Library" (Varosi *et al.* 1990) by the end of 1990. Inquiries about progress and availability should be directed to Gitta Domik[1].

The authors thank Dave Ewing, Richard Fox, Barry Hamilton, Mike Sprenger, Huang Titzi, Jeff Whaley, and other graduate students at the Electrical and Computer Engineering Department at the University of Colorado for their discussions and thoughts relating to this paper. G. Domik's efforts to develop the algorithms were sponsored in part through NASA grant NAGW-1902; efforts to document and test the programs are being sponsored by NASA grant NAGS-1390.

REFERENCES

Beichman, C. A., Neugebauer, G., Habing, H. J., Clegg, P. E., and Chester, T. J., eds. 1985, *Infrared Astronomy Satellite (IRAS), Catalogs and Atlases*, Explanatory Supplement (Pasadena: Jet Propulsion Lab).

Domik, G. O., and Brugel, E. W. 1990, *Detection of Herbig-Haro Objects in the Infrared*, in preparation.

Gonzalez, R. C., and Wintz, P. 1987, *Digital Image Processing* (Reading, MA: Addison-Wesley Publishing Company, Inc.).

Kennealy, J. P., Korte, R. M., Gonsalves, R. A., Lyons, T. D., Price, S. D., LeVan, P. D., and Aumann, H. H. 1987, *SPIE*, 804, .

Saken, J. M., Shull, M., Fesen, R. 1990, *An IRAS Survey of Galactic Supernova Remnants*, in preparation.

[1] Internet address: DOMIK%COSMOS@VAXF.COLORADO.EDU, SPAN:CYGNUS::DOMIK;bitnet:DOMIK%COSMOS@COLORADO.BITNET.

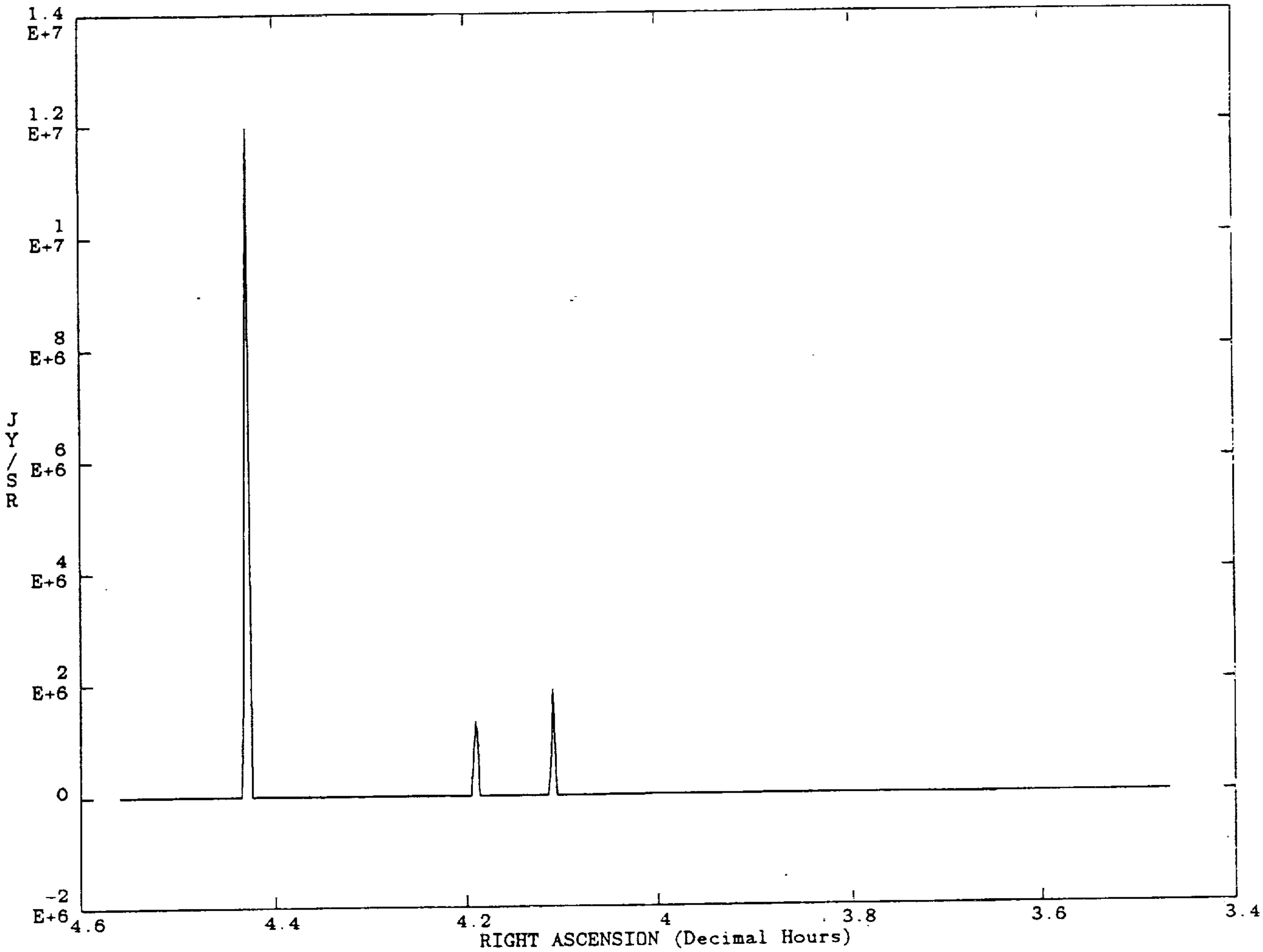


FIG. 2–Profile at a declination of 9°59′ of sky-flux plate 75, HCON 3, band 2, after flattening background. (c) Point sources in the same profile line.

Stern, D. 1984, *Interactive Software for Image Processing, Electronic Imaging*, March 1984.

Van Buren, D. 1987, *A.J.*, 94, 1092.

Varosi, F., Landsman, W. B., and Pfarr, B. 1990, *The IDL Astronomy User's Library*, 176th AAS Meeting, June 1990, Albuquerque, New Mexico.

Whaley, J. 1989, *Destriping IRAS Images*, Final project report, Fall 1989, in "Digital Image Processing", University of Colorado, Boulder.

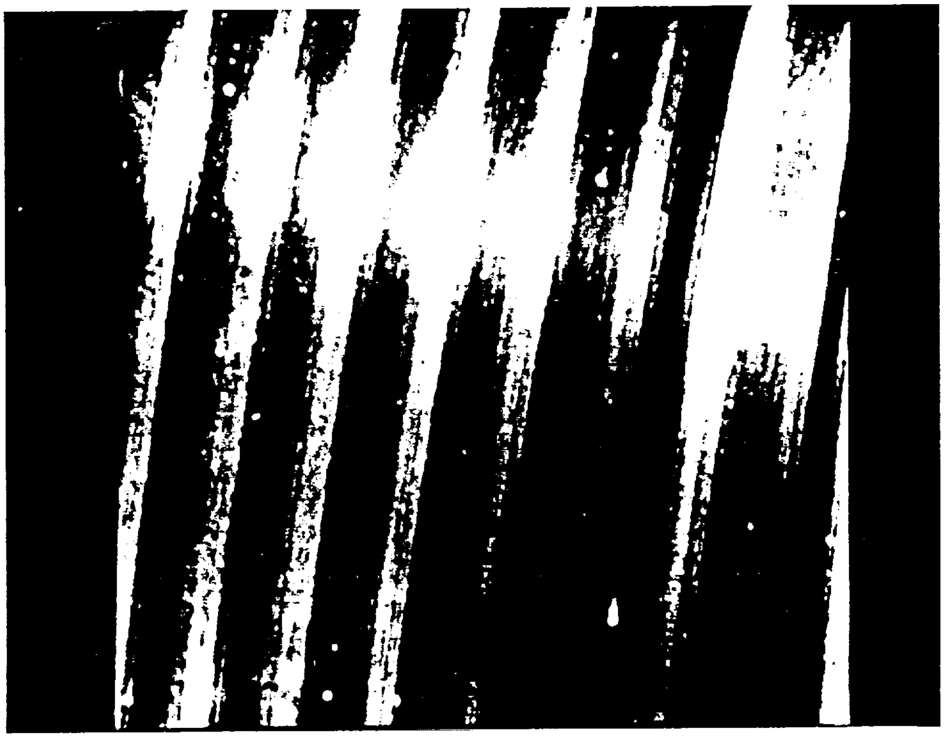


FIG. 3–Sky-flux plate 75, HCON 3, band 2. Right ascension is 4 hours, declination 15° at center of plate. Sky-flux image has been zoomed in to improve visual identification of degradations and enhancements. (a) Flattened background only; most stripes show up at an angle between 7° and 13°.

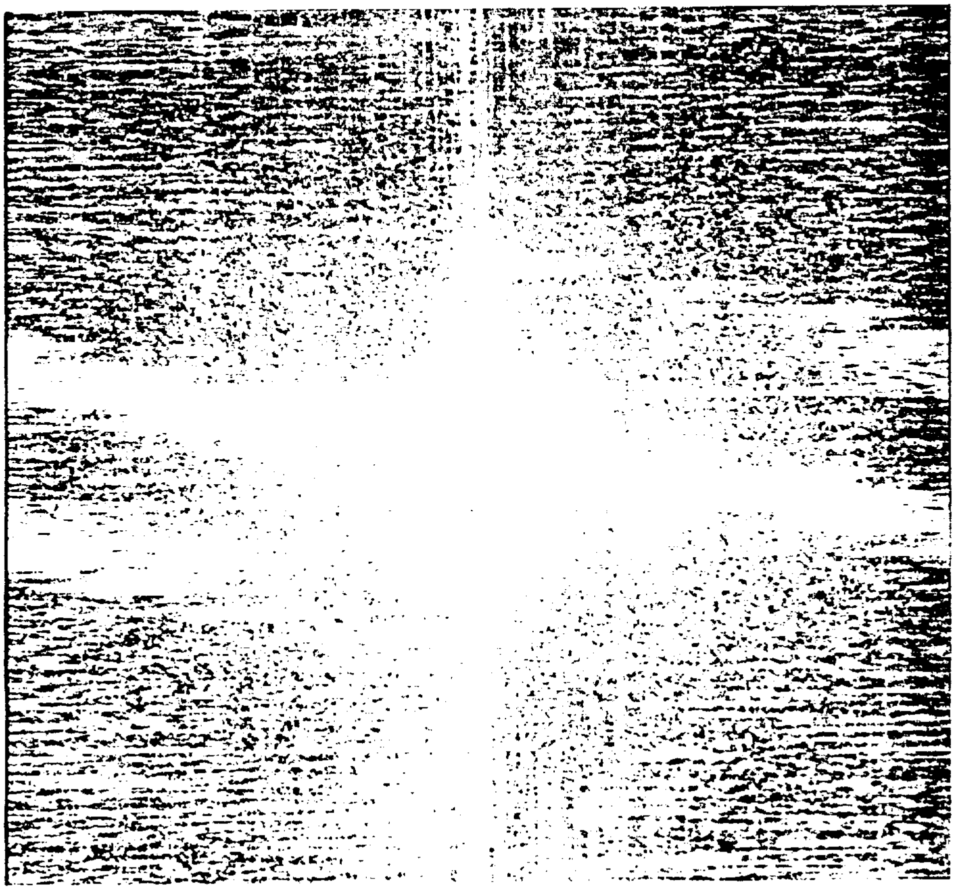


FIG. 3–Sky-flux plate 75, HCON 3, band 2. Right ascension is 4 hours, declination 15° at center of plate. Sky-flux image has been zoomed in to improve visual identification of degradations and enhancements. (b) Transformation of background image to the frequency domain: Stripes appear orthogonal to original image in the form of a series of impulses.

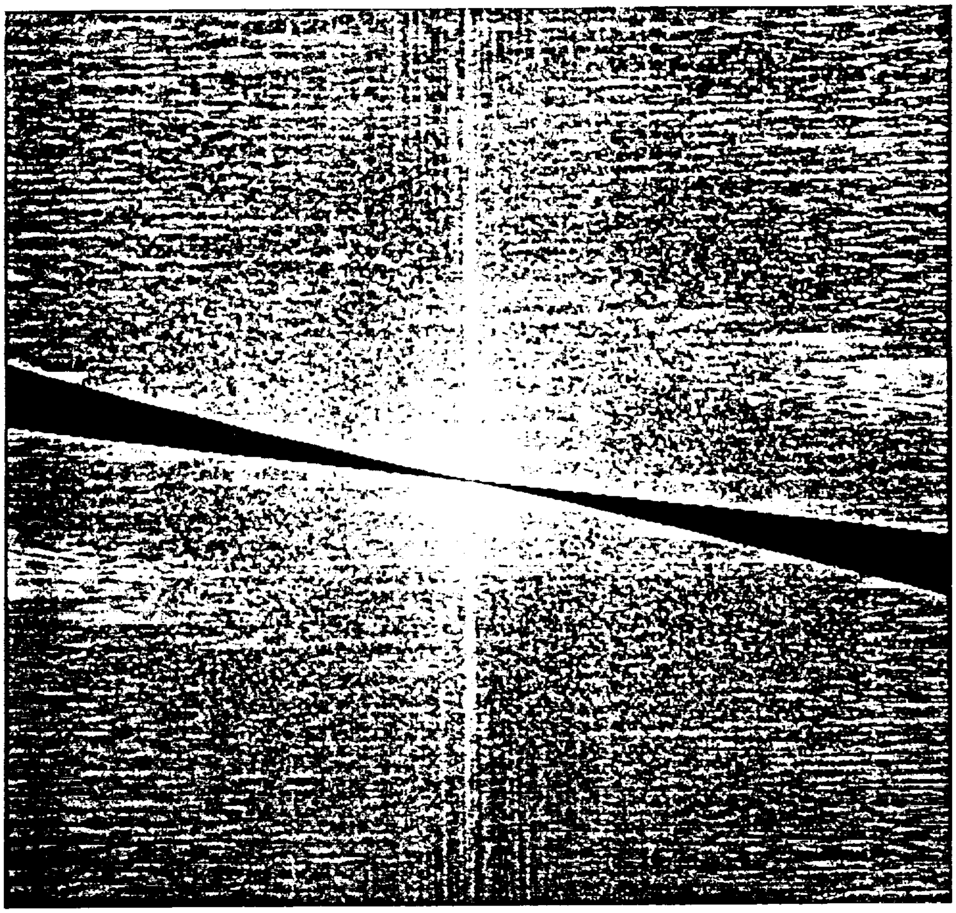


FIG. 3–Sky-flux plate 75, HCON 3, band 2. Right ascension is 4 hours, declination 15° at center of plate. Sky-flux image has been zoomed in to improve visual identification of degradations and enhancements. (c) Fourier spectrum of image with wedges shown over affected areas.

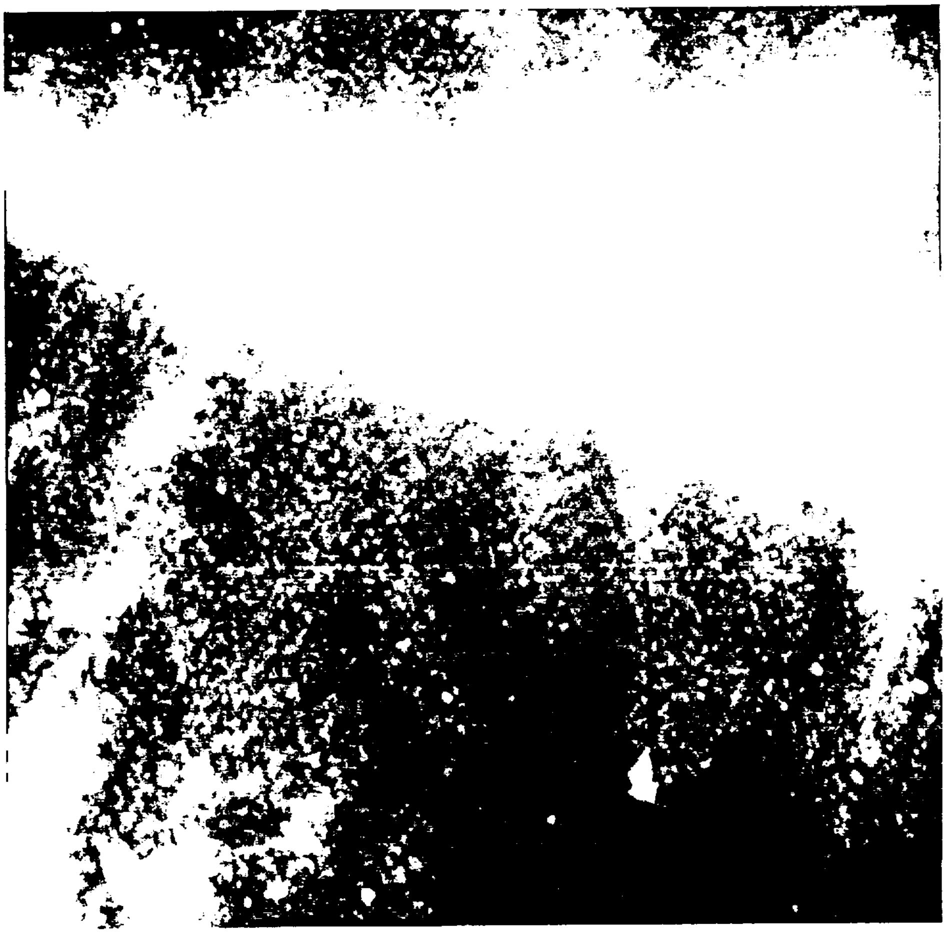


FIG. 3–Sky-flux plate 75, HCON 3, band 2. Right ascension is 4 hours, declination 15° at center of plate. Sky-flux image has been zoomed in to improve visual identification of degradations and enhancements. (d) After filtering, the sky-flux image shows strongly reduced stripes.

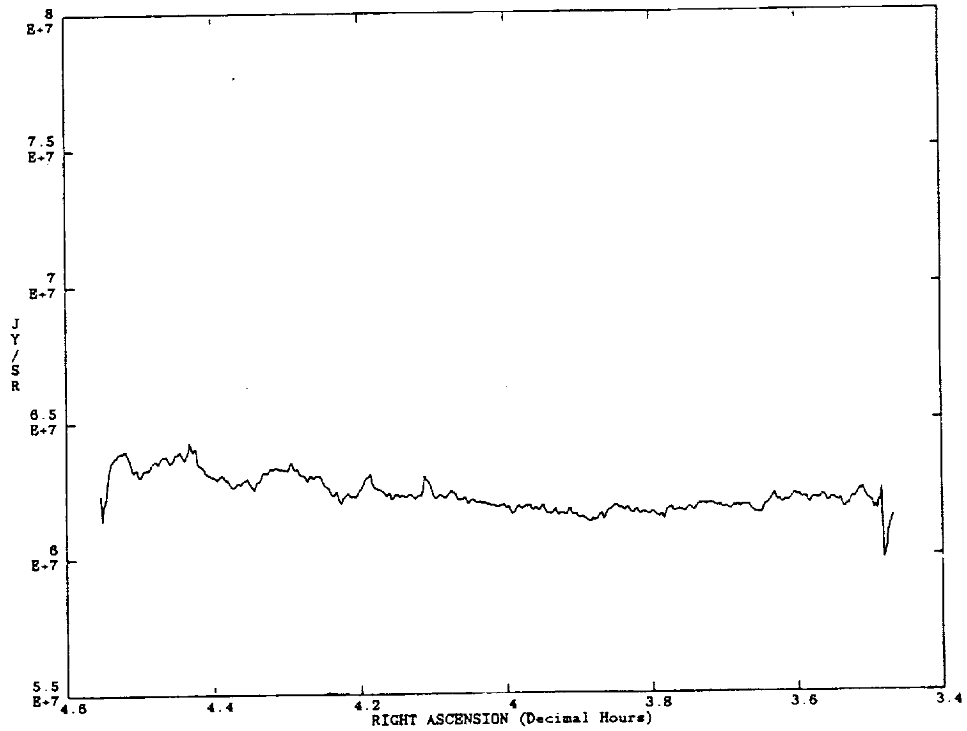


FIG. 4–Profile at a declination of 9°59′ at sky-flux plate 75, HCON 3, band 2. (a) After destriping, but before adding point sources (compare to profile line in Fig. 2(b)).

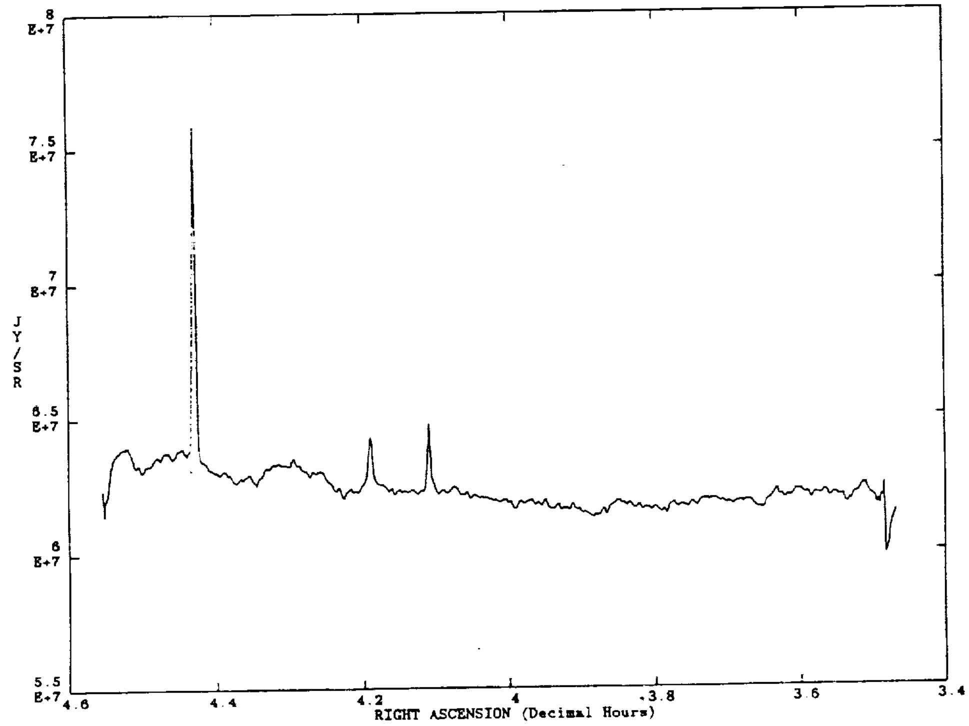


FIG. 4–Profile at a declination of 9°59′ at sky-flux plate 75, HCON 3, band 2. (b) After adding point sources (compare to profile line in Fig. 2(a)). Even though the periodic degradations are eliminated, the information of the sky-flux images seem fully contained.

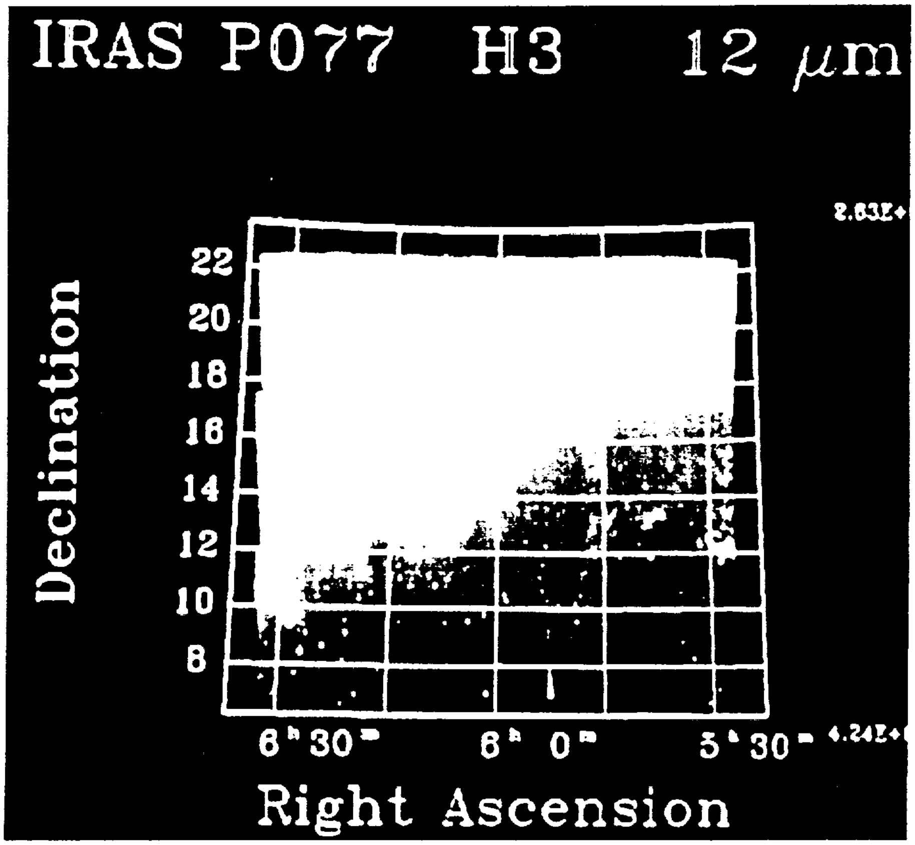


FIG. 5(a)–Original sky-flux image, plate 77, HCON 3, band 1.

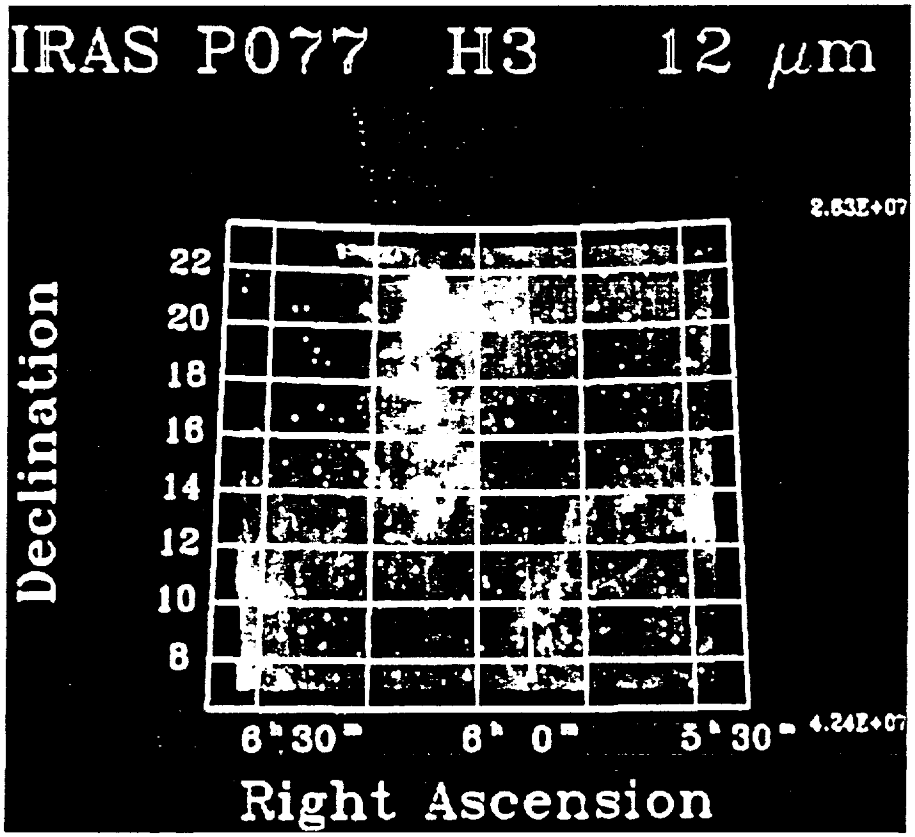


FIG. 5(b)–Plate 77, HCON 3, band 1 after flattening and destriping.

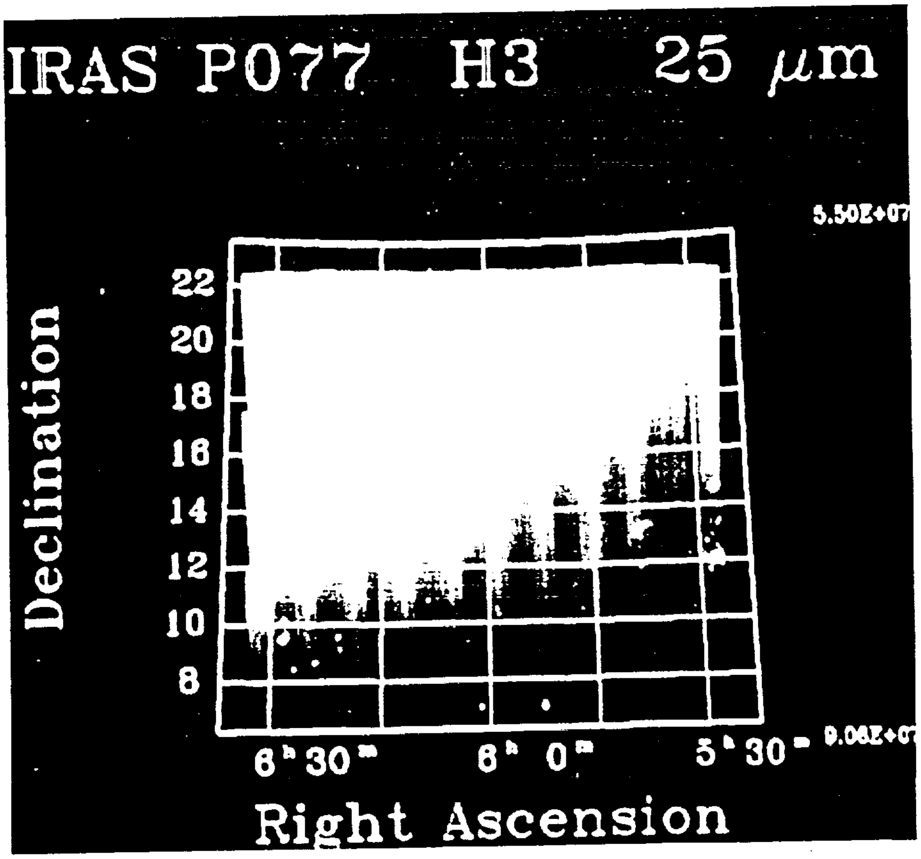


FIG. 5(c)–Original sky-flux image, plate 77, HCON 3, band 2.

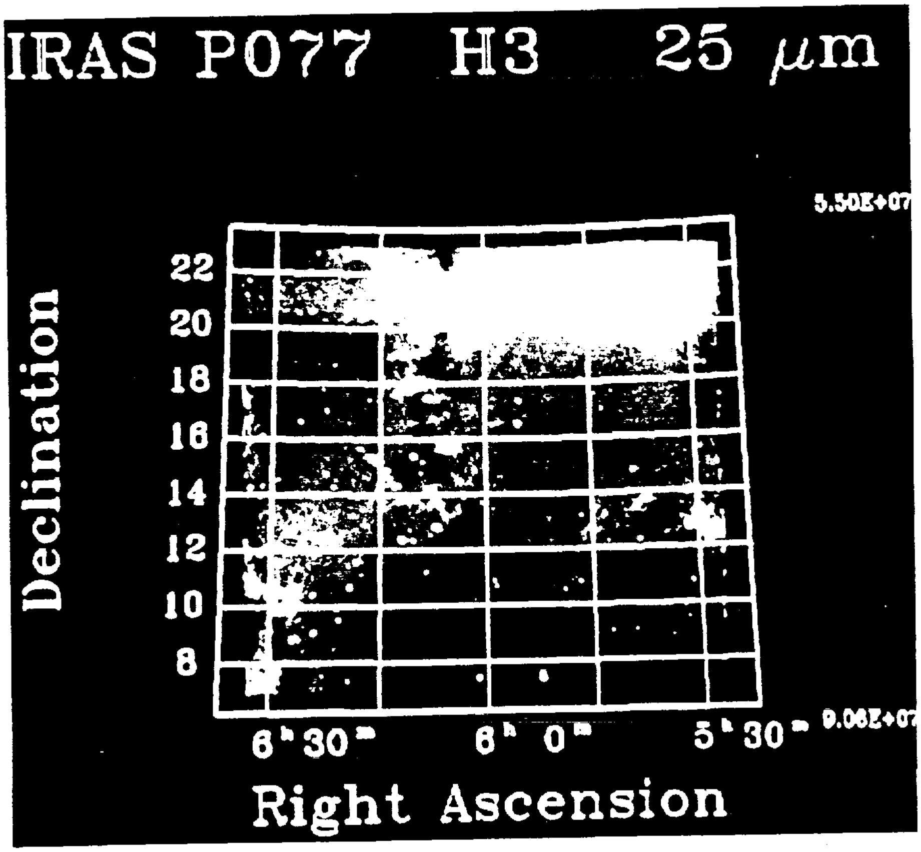


FIG. 5(d)–Plate 77, HCON 3, band 2 after flattening and destriping.

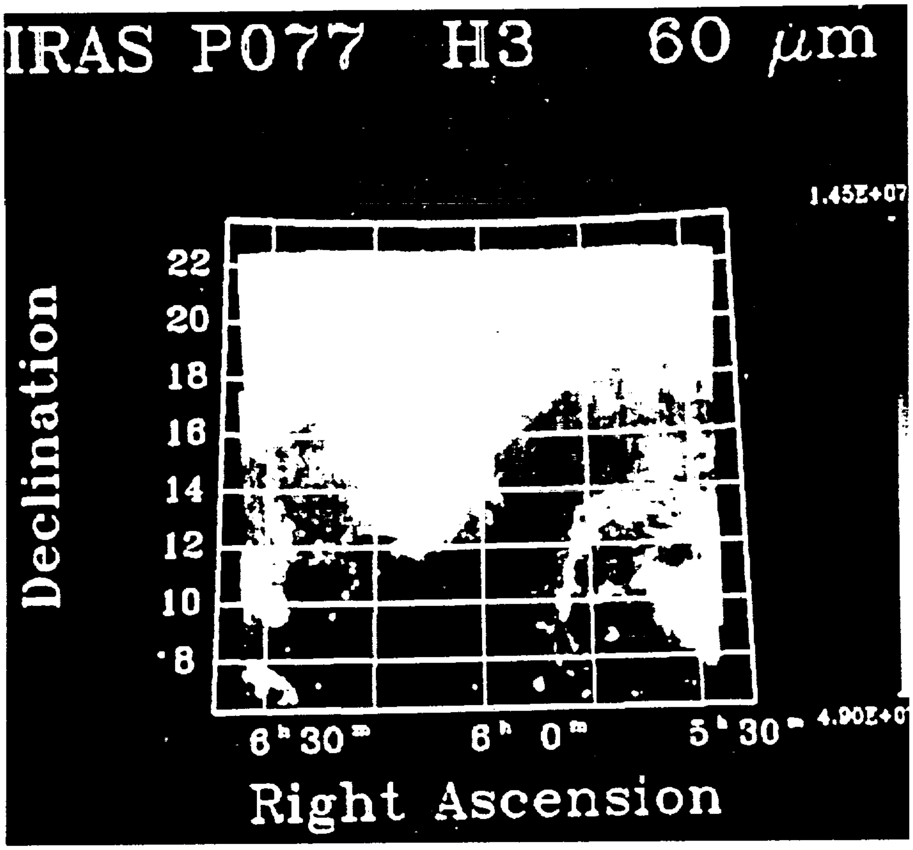


FIG. 5(e)–Original sky-flux image, plate 77, HCON 3, band 3.

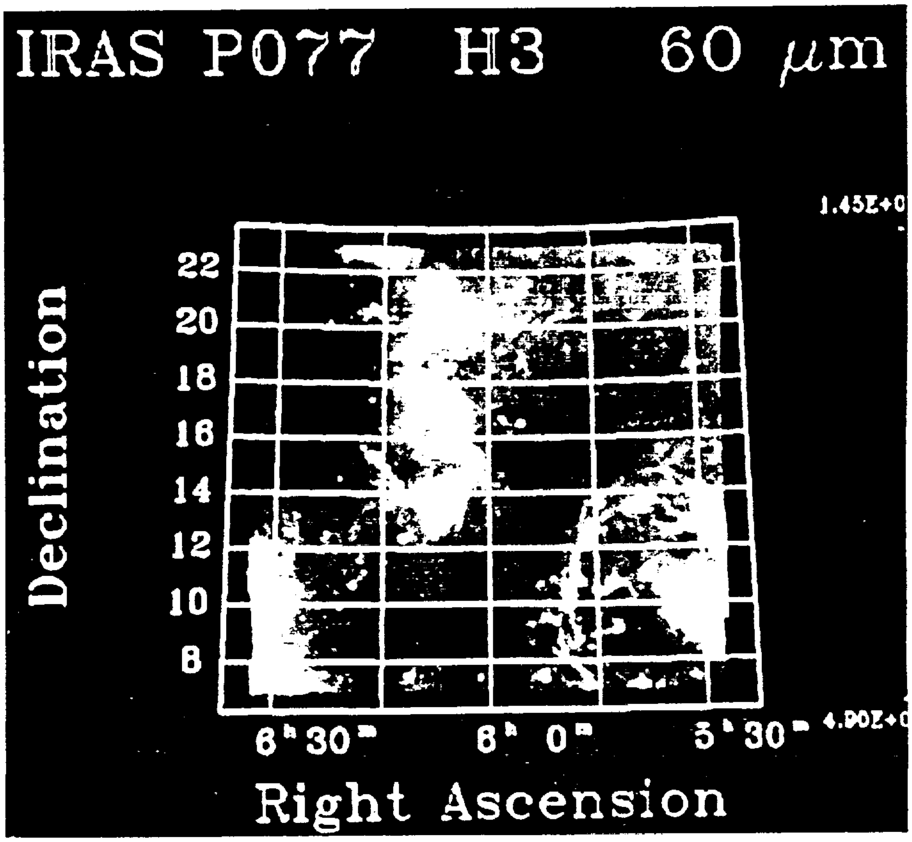


FIG. 5(f)–Plate 77, HCON 3, band 3 after flattening. No destriping necessary.

Appendix B:

IDL Tools for Volumetric Data Analysis

IDL Tools for Volumetric Data Analysis*

by
Paul Pinkney
Department of Computer Science
University of Colorado

* Supported by NASA/Astrophysics Division, NAGW-1902

1. Introduction

The Interactive Data Language (IDL) developed by Research Systems, Inc. is a powerful tool for displaying scientific data in a variety of formats. As a part of this software package, the IDL slicer was developed. Using the slicer, a user can view three-dimensional data using a subset of the IDL graphical rendering routines with the benefits of a Graphical User Interface (GUI). Some of the data analysis tools include those to render isosurfaces, data blocks, and orthogonal data slices.

In addition to IDL's data slicer, a series of IDL routines was developed to handle astrophysical data stored in the Flexible Image Transport System (FITS) format. These routines include those that allow the user to view FITS data in the IDL slicer, filter FITS data, and quantify structures within FITS data. All of these IDL routines will be described in detail in the following sections.

2. Getting Started

To use the IDL extensions, the user must first enter the IDL software package. For information on obtaining or using IDL, see your systems manager.

To enter IDL, simply enter **idl** in the login shell, at which time the user should see something like the following.

```
IDL. Version 2.2.2
Copyright 1989-1992, Research Systems, Inc.
All rights reserved. Unauthorized reproduction prohibited.

IDL>
```

As a convention throughout this document, words in a bold character format will represent user entries.

3. Viewing Data Stored in FITS Format

Many FITS data sets are available as a series of two-dimensional data slabs. To produce a three-dimensional volume of data which may be used in the IDL slicer, they must be loaded sequentially and put into a single three-dimensional array. The slicer can then render isosurfaces, slices, or data blocks within this data volume. To read in this type of FITS data set and enter the IDL slicer, the routine *fits_slicer* was developed. This routine allows the user to select which FITS data set is to be used in the extended IDL slicer.

3.1. Command Line Interface

fits_slicer, data_set, start, finish, $x = [x_1, x_2], y = [y_1, y_2]$

Positional Arguments

data_set	Name of FITS data set; character string
start	Extension of first FITS data file to be read; integer
finish	Extension of last FITS data file to be read; integer

Optional Keyword Arguments

x	Computational window for data in first dimension, integer vector
y	Computational window for data in second dimension, integer vector

This routine reads FITS data into a single three-dimensional array. The first argument, *data_set*, is a character string that defines which data set is to be read. If this data set is not in the user's current working directory, either a relative or absolute path to the data set must be given. For example, if the data set *12co* resides in the directory */images/L1551*, then *data_set* must be **'/images/L1551/12co'**.

The arguments *start* and *finish* are the first and last extensions of the two-dimensional slabs of FITS data to be read. Continuing with the above example, if the files *12co.1*, *12co.2*, ..., *12co.25* are in the directory */images/L1551*, and the user wished to read the entire sequence of files, then *start* and *finish* would be **1** and **25**, respectively. The two-dimensional slabs of data would be placed sequentially in a three-dimensional array as shown in Figure 1. For many data sets stored in FITS format, the x- and y-axes are spatial dimensions, but the z-axis, in general, will not be a spatial dimension. Instead, it may imply a direction of increasing velocity, for example.

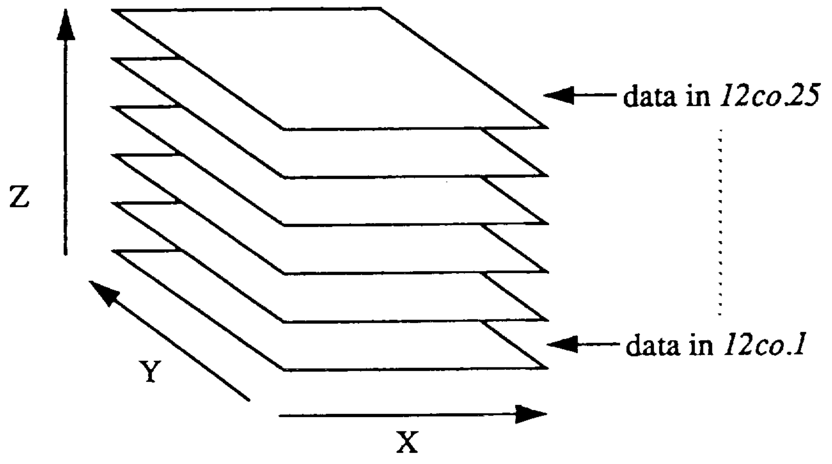


Figure 1. Organization of FITS Slabs in Data Volume

Optionally, a subset of the horizontal domain may be viewed with the slicer instead of the full domain. Specifying a windowed domain is done through the use of the keyword arguments *x* and *y*. These arguments are IDL vectors which define the domain window in horizontal computational space; that is, they specify beginning and ending indices in the first and second dimensions. For example, consider a data set in which the two-dimensional slabs of data are 200 by 400. If a 100 by 100 portion at the center of the data set was to be viewed with the slicer, then the user could enter **[50, 150]** for *x* and **[150, 250]** for *y*.

Completing this example, the full command entered at the IDL prompt would be

```
IDL> fits_slicer, '/images/L1551/12co', 1, 25, x = [50, 150], y = [150, 250]
```

3.2. Graphical User Interface

The original IDL slicer is described in *IDL Basics* by Research Systems, Inc. This slicer was modified for both the general user and for those who analyze FITS data. These enhancements are discussed in the following sections.

3.2.1. Rendering Data Slices

The original slicer was limited to displaying orthogonal data slices, that is, cross sections that were perpendicular to one of the coordinate axes. A new interface was developed to allow arbitrary rectangular slices to be rendered.

The cross sections of data that may be viewed with the extended slicer are based on a rotational axis. This axis is a single line within the slice to be displayed. Instead of the entire slice being perpendicular to a coordinate axis, the rotational axis is parallel to an axis. Three parameters for a slice of data may be specified:

- The orientation of the rotational axis
- The position of the rotational axis
- The angle of the slice about the rotational axis

Finally, a button is pressed to signal the program to display the slice. The entire interface is explained in the following sections.

3.2.1.1. Orientation

The orientation of the rotational axis is specified by pressing the right mouse button until the desired orientation appears on the interface. The three possible orientations are illustrated in Figure 2, the rotational axis being in the center of the plane that identifies the orientation.

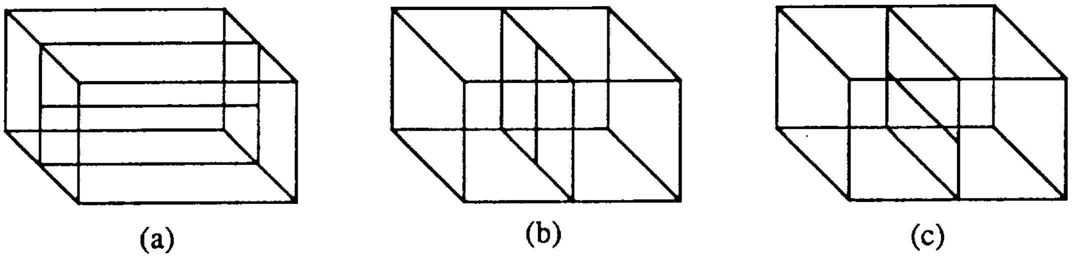


Figure 2. Extended Slicer Interface Orientations

Figures 2(a), (b), and (c) represent the rotational axis being parallel to the x-, y-, an z-axis, respectively. In the actual interface, the orientation plane is gridded, but these grid divisions are not represented to simplify the diagrams.

3.2.1.2. Position

The position of the rotational axis may be moved anywhere within the domain of the other two coordinates. For example, if the user chooses to have the rotational axis parallel to the x-axis, it may then be given any (y, z) location within the domain of the three-dimensional volume. Figure 3 shows how the rotational axis may be positioned within the data volume.

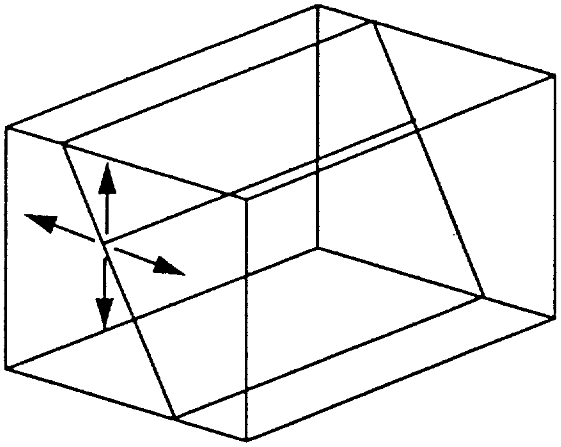


Figure 3. Selecting a Data Slice Angle Using the Extended Slicer Interface

3.2.1.3. Angle

The last parameter that may be specified when selecting a slice is an angle. This angle is that of the data slice rotated around the rotational axis. Figure 4 shows how this angle is specified.

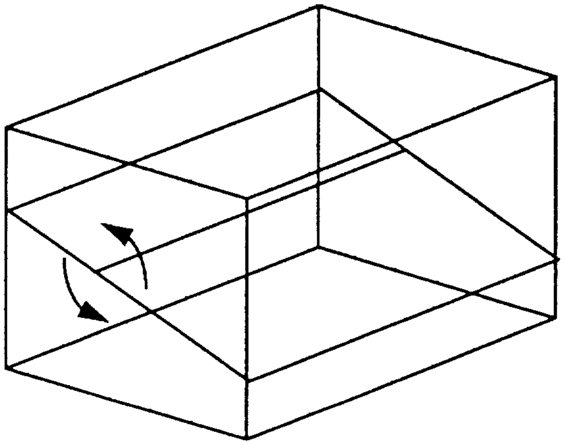


Figure 4. Selecting a Data Slice Angle Using the Extended Slicer Interface

As is done when selecting the rotational axis position, the user clicks in the foward-facing plane that is perpendicular to the rotational axis, but in this case, with the middle mouse button. The wireframe plane that represents the selected slice of data may then be rotated around the rotational axis. Once the desired angle is chosen, the mouse button is released.

3.2.1.4. Rendering the Slice

The orientation, position, and angle parameters have been specified to the user's liking, the button labelled *Render Slice* is then pressed to render the data slice. The slice will then be rendered as a shaded cross section in the large graphical window.

3.2.2. Saving Data Slices

Once a data slice is rendered, it may be saved to a file as a two-dimensional FITS data slab. Saving the slice is done by using the interface shown in Figure 5.

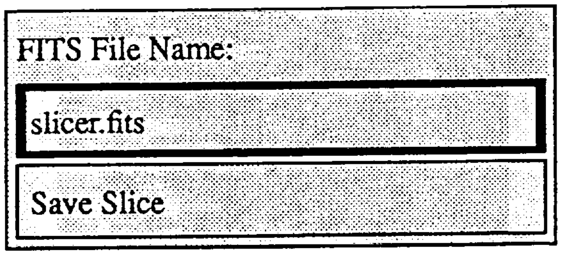
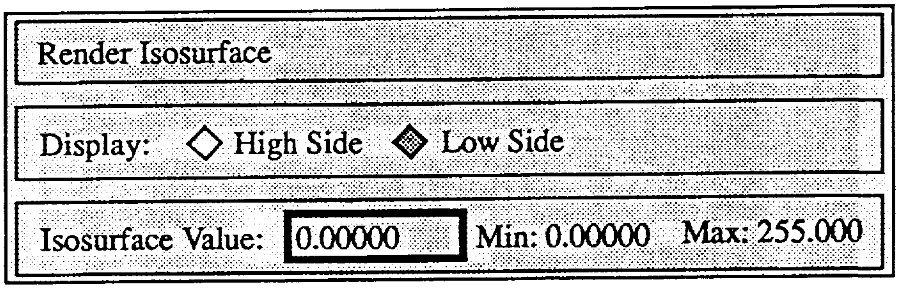

FITS File Name: slicer.fits

Figure 5. Interface for Saving Data Slices

Using the IDL text bar, enter the name of the file in which the data is to be saved. The defaults file name is *slicer.fits*. Once the file name is entered, press the button labelled *Save Slice*.

3.2.3. Rendering Isosurfaces

By selecting *Isosurface* on the main slicer interface, the user may view isosurfaces within the data volume. The interface that allows the user to enter the isosurface value is shown in Figure 6. Note that this interface differs from that in the original IDL slicer.



Display: ◇ High Side ◆ Low Side

Isosurface Value: 0.00000 Min: 0.00000 Max: 255.000

Figure 6. Interface for Rendering Isosurfaces

Again, a text bar is used. A value between the displayed minimum and maximum values may be entered. The user also may select whether the isosurface is to enclose values higher or lower than the threshold. If the isosurface is to enclose values lower than the threshold, press the button labelled *High Side*; otherwise, press the button labelled *Low Side*. Finally, pressing *Render Isosurface* displays the isosurface in the large graphical window.

3.3. Related Routines

The routine *fits_slicer* uses two other routines that may be of interest. The first routine is that which reads in the sequence of FITS slabs. Second, the extended IDL slicer is the GUI that allows the user to render the data volume in various formats. The following sections describe these routines.

3.3.1. Reading FITS Data

The function *read_seq* reads in a sequence of FITS data slabs and puts them into a common three-dimensional volume.

volume = read_seq (data_set, start, finish)

Arguments

data_set	Name of data set; character string
start	Extension of first FITS data file to be read; integer
finish	Extension of last FITS data file to be read; integer

The arguments for this routine are the same as *data_set*, *start*, and *finish* for *fits_slicer*. The data slabs are placed in the three-dimensional array in the manner illustrated in Figure 1. The resulting data volume is returned in *volume*.

3.3.2. Rendering Data

The procedure *arb_slicer* is an extension of the original IDL slicer.

arb_slicer, volume

Arguments

volume	Volume to be used in the slicer; three-dimensional array

If the user has a three-dimensional volume of data in *volume*, it may be fed directly to the extended IDL slicer using this interface.

4. Filtering FITS Data

To filter values out of a sequence of FITS data slabs, use the routine *fits_filter*. This routine loads a sequence of two-dimensional FITS slabs into a single three-dimensional data array in the same manner as *fits_slicer*. Data values either above or below a given threshold are filtered out of the data volume and written to disk as a three-dimensional FITS data array.

4.1. Interface

fits_filter, data_set, start, finish, thres, new_file, high = high, low = low

Positional Arguments

data_set	Name of data set; character string
start	Extension of first FITS data file to be read; integer
finish	Extension of last FITS data file to be read; integer
thres	Threshold value for data filtering; same type as data volume
new_file	Name of file for filtered data; character string

Mutually Exclusive Keyword Arguments

high Specifies values higher than *thres* to be filtered out of data; flag
low Specifies values lower than *thres* to be filtered out of data; flag

4.2. Description

The argument *data_set*, *start*, and *finish* are identical to those in the routine *fits_slicer.*

Arguments *high* and *low* are mutually exclusive; that is, either one or the other of them is specified, but not both. These flags determine how the data is filtered. If the user specifies **/high**, then values higher than the given threshold value are filtered from the data. Similarly, if **/low** is specified, values lower than the threshold are filtered. The argument *thres* is the threshold value. All values filtered out of the volume are replaced by *thres* in the resulting data array.

Once the data is loaded into a single three-dimensional array and is filtered, the resulting data volume is written to the file specified by *new_file.*

Using the sample data set from Section 3.1, consider the following example. If values above 2.1 were to be filtered out of the data and the resulting data volume were to be written to the file *myfile.dat*, then the user would enter

```
IDL> fits_filter, '/images/L1551/12co', 1, 25, 2.1, 'myfile.dat', /high
```

5. Quantifying FITS Objects

Displaying an isosurface from a three-dimensional data volume can provide information about the behavior or structure of a system, but often a more quantified approach is necessary. For this purpose, the routine *fits_objects* was created. This routine sums the values inside each individual object defined by an isosurface.

5.1. Interface

fits_objects, data_set, start, finish, thres, $x = [x_1, x_2]$, $y = [y_1, y_2]$, *low = low, high = high*

Positional Arguments

data_set Name of data set; character string
start Extension of first FITS data file to be read; integer
finish Extension of last FITS data file to be read; integer
thres Bounding value for data objects; same type as data volume

Required Mutually Exclusive Keyword Arguments

low Specifies that objects are defined as those structures bounded by and enclosing values lower than *thres*; flag
high Specifies that objects are defined as those structures bounded by and enclosing values higher than *thres*; flag

Optional Keyword Arguments

x	Computational window for data in first dimension; integer vector
y	Computational window for data in second dimension; integer vector

5.2. Description

When an isosurface is rendered, the structures that are displayed are those bounded by a threshold value and containing values either higher or lower than that value. The routine *fits_objects* is used to add up the values inside each of these structures.

The arguments *data_set*, *start*, and *finish* are the same as those in the routines *fits_slicer* and *fits_-filter*. The argument *thres* is of the same data type as that of the three-dimensional data volume and specifies the bounding value for the data objects in the same way a threshold value is given for isosurfaces. Either values above or below this value will be contained in the data objects, depending on whether **/high** or **/low** is specified.

Domain windowing may be done using the arguments *x* and *y*. These arguments are identical to those in *fits_slicer*.

As an example, suppose the following was entered:

```
IDL> fits_slicer, '/images/L1551/12co', 1, 25
```

Notice that variables *x* and *y* were not specified, so the entire data volume will be used in the extended slicer.

Using the *Isosurface* option in the slicer, the user then obtained the isosurface shown in Figure 7. This isosurface encloses values higher than 2.0. Notice that two distinct objects are revealed: a large structure and a much smaller one. If the user entered

```
IDL> fits_objects, '/images/L1551/12co', 1, 25, 2.0, /high
```

the output shown in Figure 8 would be displayed. In this figure, three objects were identified, but one of these contained a single value within the data volume. The other two objects are those displayed in Figure 7, but revealed in a quantified format.

ORIGINAL PAGE IS
OF POOR QUALITY

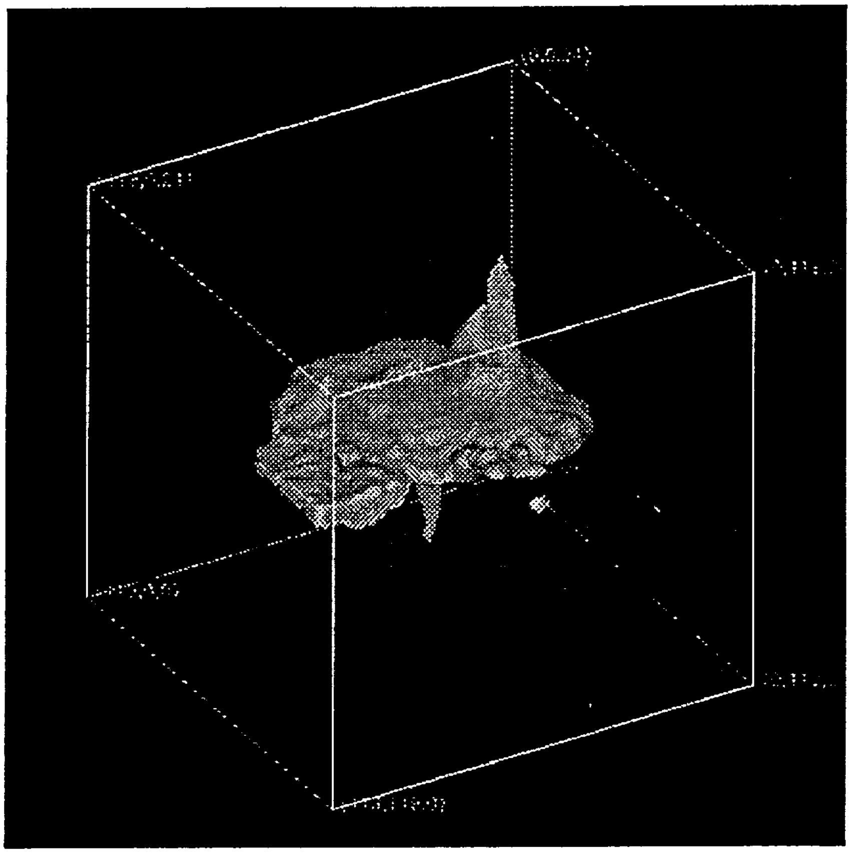


Figure 7. Graphical Representation of L1551 Isosurface Enclosing Values Greater Than 2.0

```
Objects enclosing values higher than 2.00000

Object 1: 45004.1
Object 2: 2.00966
Object 3: 46.4203

Total number of objects: 3
Total sum for all objects: 45052.5
```

Figure 8. Quantified Representation of L1551 Isosurface Enclosing Values Greater Than 2.0

6. Conclusion

In conclusion, it is the hope of everyone involved in the development of these tools that they become quite useful in the analysis of astrophysical data sets as well as other types of three-dimensional data. Experience has shown that experimentation with these tools is the best way to reveal their usefulness. Play a little while and see what is discovered.

Appendix C:

Potential Research Aspects of 3-D User Interfaces

Potential Research Aspects of 3D User-Interfaces*

Salim Alam

November 12, 1992

Contents

*Supported by NASA/Astrophysics Division, NAGW-1902

1 Introduction

This work has been motivated by a lack of commonly available, low cost and effective user-interfaces for three-dimensional interaction, especially the manipulation of large volumetric data-sets. Workstations which allow fast access to three dimensional graphics have become increasingly common and are being used for a wide variety of tasks, from solid modeling to medical imaging to animation [1]. However, while the hardware has become increasingly sophisticated, the user-interface has generally retained its two-dimensional heritage — a number of commercial programs are available which provide very advanced methods of visualizing data but restrict the user to interacting with the data in only certain fixed ways, which are ultimately unintuitive.

Although high-end research institutions have been working for a long time on problems of user-interface design for three-dimensions, the state-of-the-art interfaces have been custom-designed for certain applications and are also very expensive. However, this is starting to change — sophisticated three-dimensional user-interface hardware is coming into a price range that is affordable by the average user.

With this work we hope to highlight some of the problems with current implementations, look at a number of state-of-the-art hardware and software solutions and attempt to point out some of the promising new tools and paradigms. This may then be resolved into a system that provides a better cost/performance ratio. Of course, the measurement of "performance" needs to be defined in terms of some explicit criteria.

2 Defining the Problem

Experience with some of the commonly available visualization tools shows areas of deficiency that could be improved. Below, we describe some of these deficiencies.

IDL Slicer. IDL is a popular interactive language for visual data analysis. The IDL system also provides some built-in tools for three-dimensional visualization. One of these tools that we frequently use is the *data slicer.*

The data slicer allows one to visually examine volumetric data in a variety of ways [2]. The data can be examined in any orthogonal plane. Furthermore, a subsection or "sub-cube" can be selected and examined. A simple three-button mouse is used to interact with the slicer.

The interface for selecting the sub-cube, while technically competent, is cumbersome to use because it is difficult for the user to manipulate the cursor in three dimensions – the visual cues are not very strong. Furthermore, it is difficult to select and pinpoint the exact area that one wants to examine because of the un-intuitive selection method.

AVS Geometry Viewer. Another very powerful tool that we have been using is the Application Visualization System (AVS). The Geometry Viewer allows rotation and translation of objects in an arbitrary manner, using a combination of mouse and keyboard keys. While this is an excellent tool for the most part, we have found that when precise positioning of complicated three-dimensional objects is needed, the rotation and translation methods become very difficult to use. For example, on one occasion it took upwards of twenty minutes to precisely position two six-atom molecules using the mouse controls.

While it is possible to use a Spaceball three-dimensional input device to interact with AVS [3], there is anecdotal evidence [1] that suggests that the precise positioning and rotation still remains a cumbersome process. Part of the problem stems from the fact that the display is monoscopic.

It is clear that these user-interface deficiencies are directly related to the use of two-dimensional input and output devices, and can be grouped into three different areas:

- **Positioning.** This means that it is difficult to precisely position an object in 3D space. The main reason is the difficulty of accurately perceiving where in space the cursor actually is.
- **Rotation.** Using mouse and keyboard areas to interact with an object is cumbersome and not intuitive. Furthermore, it forces the user to remember complex mouse and keyboard sequences.

[1] Replies received from queries posted on Usenet *alt.3d* newsgroup.

- **Selection.** Selecting an area of the 3D space to examine further is probably the most difficult of all, since it requires both accurate positioning and complicated selection commands to remember. Moreover, the depth cues are not sophisticated enough to sustain accurate positioning.

We need to look at ways to eliminate or minimize the deficiencies. New user-interface hardware that is effective and affordable is becoming increasingly common, and there is also continuing research in new software paradigms for user-interfaces.

3 State-of-the-Art Interfaces

Cutting-edge research in user-interface design seems to have a number of things in common. Firstly, the user-interface is usually custom-designed for a specific application. Secondly the main component is designed to take advantage of a very specific technological break-through — for example, holography. And thirdly, the cost of such a system is quite high since most of the technology is not off-the-shelf.

The following subsections describe some important research contributions in three separate areas of user-interface design.

3.1 Input Devices

A three-dimensional input device is a six-degree-of-freedom device that can sense position as well as orientation. Usually one needs special hardware to track the position of the user's physical cursor (such as the hand or a stylus). The most commonly used tracking device is manufactured by Polhemus and uses a magnetic field to locate the sensor.

The Bat. The "Bat" is a tool designed by Colin Ware and Danny Jessome at the University of New Brunswick [4]. It is basically a three-dimensional mouse, constructed using a Polhemus device. The mouse is actually a sensor that senses a signal output from a fixed source. A controller measures the position of the sensor and communicates with a host computer.

The effectiveness of the bat as an input device was tested using 3D placement as the primary operation. Two important factors that play a role in accurate operation were found to be kinetic depth (ie, a feeling of 3D when object is moved) and a simple protocol.

3-Draw. 3-Draw is a tool for designing three-dimensional figures, specifically for CAD. It was constructed as part of a research project at MIT [5]. It uses two six-degree-of-freedom sensors, both of which are part of a Polhemus 3Space Tracker. One sensor is attached to a plate and is held in one hand, while the other is attached to a stylus and is held in the other hand. Using two sensors allows the user to use one of them to function as a plane in the virtual world, while the other is used to draw three-dimensional points relative to the plane. Software was written that interactively reflects the position of both the sensors as well as the plotted points.

The effectiveness of the tool was measured by making CAD drawings. It was found that working directly in three dimensions was much more faster and natural.

3.2 Output Devices

Some of the problems with having a 2D input device can be alleviated by using a three-dimensional output device. Research has indicated that for simple tasks such as positioning an object with a mouse, a stereoscopic display cuts down on the time needed for accurate positioning of the object [6]. It is not unreasonable to extend this thesis by claiming that in general for these kinds of tasks any 3D output device can make it easier to interact with a data-set. More research is needed, however, to ascertain exactly what the benefits are for various devices.

We discuss two state-of-the-art output methods below.

Stereoscopic Monitors. Stereo monitors allow the user to observe a scene in three dimensions. They work by transmitting two different images, one to each eye. This parallax leads to a 3D effect.

There are many different ways that monitors send different images to each eye. Some monitors use special shuttered glasses which automatically keep in sync with the picture being transmitted — while the scene for one eye is being displayed the shutters for the other eye are closed, and vice versa. Monitors made by Stereographics use this type of technology [7]. Other monitors place the shuttering mechanism directly on the monitor, and require the user to wear simple passive polarized glasses — monitors made by Tektronix fall into this category. [2]

[2] One testament of the usability of stereoscopic displays is that they are starting to be used in commercial ventures. For example, Vexcel Corp. in Boulder currently uses the Textronix stereographic system, along with a mouse and a dialbox for input, to create

The most recent research has resulted in auto-stereoscopic displays that do not require the user to wear any kind of glasses. Three such displays are currently available, with more models being in the research stages. The most promising of these is a model developed by Dimension Technologies, Inc. which uses a vibrating mirror to project various cross-sections of the two-dimensional image onto 3D space. Both monochrome and color versions are available, with a resolution of 320x480 pixels in 16 gray-shades or 32 colors.

Holography. While still in mostly the research state, holography offers another method of three-dimensional vision, including a "walk-around" effect through a number of degrees. Holograms are based on use of laser light, but there are a number of ways to reproduce a holographic image [8]. The most common method is film based [9], but it has serious limitations. Another method, which is being researched by engineers at Texas Instruments, is based on the projection of laser light onto a spinning helical surface which is enclosed in a dome-like structure. Finally, researchers at MIT have been working on generating holographic images using computer-controlled generation of light-patterns which are focused by optics to generate real-time holographic images — this process is called *holovideo.* These holograms however require enormous computing power and storage, and are still limited by a viewing angle of only 15 degrees. However, recently there have been some breakthroughs which have decreased the storage and computation needs while increasing the viewing angle [10].

3.3 Virtual Reality

Virtual Reality combines both three-dimensional input and output devices to provide a complete virtual environment for the user to manipulate. The main advantage of virtual reality is that the user is completely "enclosed" by the virtual environment and can interact with it using normal body movements. The main disadvantage of virtual reality is the difficulty of completely simulating the environment — there are many restrictions. However, as we shall see, virtual reality interfaces are being used and offer much potential for solving many of the problems associated with positioning, rotation and selection of data.

three-dimensional surface maps from satellite data. The quality of the three-dimensional images is very good – when correctly calibrated the images appear solid and flicker-free.

Two different approaches to a virtual reality environment are described below.

Virtual Wind Tunnel. In [11] researchers describe how they have developed a virtual reality environment to visualize and interact with unsteady flow simulations. The system consists of a stereo head-tracked display which is worn on the user's head using a "boom" device. The boom is manufactured by Fake Space Labs, and allows for the mounting of "two CRTs on a counter-weighed yoke attached through six joints to a base". The large degrees-of-freedom allow for comfortable movement for the users (within a limited spatial boundary) and the exact position and orientation of the user's head is transmitted by the boom directly to the host computer using a serial port.

In addition, the user interacts directly with the environment by using a VPL dataglove, which uses the Polhemus 3Space Tracker to determine the postion and orientation of the hand, and fiberoptic sensors to determine the flexing of the finger joints.

The entire hardware is attached to a Silicon Graphics Iris 380 VGX workstation with eight processors and a rating of 200 VAX MIPS. All the computation and rendering is done in real-time using this machine.

The user can interact with the simulation in a number of ways. The co-ordinate system for the data can be transformed by the user by simply making a fist and gesturing. Also, "rakes" and "seed points" can be placed by simply picking them up and dragging them as one would normally do with a real object.

Using this setup it was found to be very easy to experiment with various factors in the simulation — the user obtained an intuitive understanding of the flow field. The researchers found the interaction method to be very satisfactory.

VR on Five Dollars a Day. Randy Pausch at the University of Virginia has developed an experimental virtual-reality setup for research in devices to help severely disabled children [12]. His system consists of a 386-based IBM PC, a Polhemus 3Space Isotrak, two LCD displays, and a Mattel Powerglove. The software, including rendering libraries, has been custom-developed. This entire system costs around $5000.

Use of this system has outlined several criteria for virtual-reality user interfaces. Firstly, the time-lag for the head-tracker is very important and should be minimized. Secondly, while a stereoscopic display is not essential, a reference plane is. Also, some kind of method is needed to allow the user to escape the spatial constraints — for example a "vehicle" which can be moved rather than the user.

4 Software Research

The software interface also plays a large role in dealing with three-dimensional data. An interface might have the best hardware, but if the software is not friendly and intuitive the overall functionality will be diminished and the system will be difficult to use.

Researchers have been approaching the problem of creating a better software interface from various directions. Some of these approaches are described below.

2D Direct Manipulation. The Human Interface Group at Apple Computer, Inc. have designed and experimented with a system for moving solid-modeled three- dimensional objects using a monoscopic display and a single-button mouse [13]. From the very start, consideration was given to using the correct metaphors - for example, since people move things using hands, an icon of a hand was used as a cursor. Furthermore, while performing different tasks, the icon was changed to reflect the physical aspect of the task, for example, pushing, lifting or rotating. However it was found that users were still confused about actual directions and ranges of movement, and also about the position of the "hot-spots". To resolve these problems, bounding boxes and "narrative" handles were used to further clarify the interface.

VPL System Interface. An experimental system by VPL was evaluated for the task of visualizing and interacting with data consisting of the neuroanatomical structure of the human brain [14]. [3] While the researchers found this VR system to be "by far the best and easiest" method of interacting with the data, they did acknowledge some problems with the interface. Some of the problems were associated with the nature of the hardware, for example, limited scope of movement, problem with tripping over cords, slow update rates, etc. In terms of the software interface, the "flying" function

[3] VPL also sells a similar commercial system, and has since announced a much lower-cost product with even better performance. However, at even the amazing price of $58,000, only large R&D institutions can afford to purchase it.

provided by the software was found to be difficult to use due to the inability to control many aspects of the speed of movement. Also, it did provide the actual experience that one would associate with "flying". [4] The researchers suggest that a better metaphor for a virtual reality interface might be a "push/pull" interface where users could in-effect directly interact with the data. Another problem they found was the inability to "fly" in one direction while looking in another direction. This ability could provide a very unique method of visualization if it was available. Finally, the use of visual cues to warn the user when attempting to move out of the range of the tracking devices was also suggested.

Other Research. Simply modifying the software interface to minimize the kinds of problems discussed above could have a significant impact on its usability. For example, a technique for rapid controlled movement is suggested in [15], which could solve the problem with "flying" that was encountered with the VPL interface. Use of force-sensing and force-feedback devices could provide a better interface for tasks such as the handling of virtual objects [16]. And use of neural-nets could allow for a large library of gestures that could be customized for individual users [17],

5 Criteria for a New Interface

It is clear that the use of three-dimensional input and output devices, used both separately and together, provide a distinct advantage over two-dimensional interfaces. It should be possible to preserve this advantage while developing a three-dimensional interface that would be low-cost and easily available.

By studying the research results obtained we get an idea of what an interface for three-dimensional manipulation should be. We have seen a number of interfaces that have been optimized for a particular task. In our own problem definition, we are interested in the use of a three-dimensional cursor which can be used to place, rotate and select the data-set. There does not seem to be any reason why it would not be possible to develop a general interface that would be usable for a variety of tasks while providing excellent performance for our primary problem.

[4] Although Randy Pausch's system uses a similar metaphor, he does not specifically mention these problems. However, he uses the "vehicle" metaphor to circumvent the limited range of physical movement available.

Is virtual reality the ultimate interface? Perhaps, although a totally usable virtual reality interface cannot be expected in the near future. It seems that we might be better off identifying the basic advantages of a virtual reality interface and using low-cost and easily available hardware to simulate those advantages. For example, use of a glove to "grab" and interact with data on a normal monoscopic display would be much preferable to simply using the mouse — it would not matter too much that the user was not actually "enclosed" by a virtual reality.

In order to test the effectiveness of the interface we would like to use criteria similar to that used by McWhorter, et al. The primary quantitative criteria would be the time and accuracy in positioning, rotating and selecting an area of the data-set. A secondary qualitative criteria would be the elegance, intuitiveness and usability of the software paradigm.

6 Commonly Available Hardware

User-interface hardware is now becoming available at a much lower cost than the high-end hardware being used in state-of-the-art research projects. Off-the-shelf components that can be used without a lot of custom design work are described below. We will not describe any high-end components, for example those that have already been described in the previous sections — these are too expensive for normal use. For example, the Polhemus Tracker costs about $12,000, the BOOM device costs about $27,000 and the Tektronix stereo display costs about $8000. The devices we select below are available at a cost of $1500 or less.

6.1 Input Devices

- **SpaceBall 2003.** The SpaceBall is a "three-dimensional" input device. It is similar to a trackball, but does not move — the device senses the actual force exerted by the user and also the direction of the force, which allows the software to move or rotate objects in three dimensions. It is available from Spaceball Technologies and works with a large number of platforms, including Silicon Graphics and Dec. The approximate cost is $1500.00

- **Logitech 3D Mouse.** This is a six-degree-of-freedom sensor that is available for a number of platforms including the IBM PC and Silicon Graphics. Since it is serial-port based, it is easy to interface it to other platforms, although the driver software needs to be written. It can be used as a normal mouse or as a six-degree-of-freedom three-dimensional sensor. The cost is $1000.

- **Gyropoint 6D Mouse.** The GyroPoint is a three-dimensional mouse that uses gyroscopes and an optical-sensing interface to provide six-degrees-of-freedom. It has been announced with a projected release date of first quarter of 1993, and a price of $1000 for the developer version. Versions will be provided for both IBM PCs and Apple Macintosh.

6.2 Output Devices

- **3DTV Stereoscope.** The Stereoscope hardware consists of a hardware card and a pair of LCD shutter glasses for about $450. A software development kit is available for $350. For use in a virtual-reality environment these are also available with Headband, Eyeglass or Wireless models.

- **Haitex X-Specs.** These are manufactured by Haitex Resources, and sell for about $80. Using these LCD glasses and the proper software gives a stereoscopic display of computer-generated graphics. A generic interface is provided which can be used with a variety of platforms like the IBM PC or the Macintosh.

- **Sega 3D Glasses.** This is a discontinued but widely available item, available for less than $100. These provide simple stereoscopic display using any monitor. They are based on the shutter concept. Public domain serial-port interface circuits are available, as well as driver and user-interface software for the IBM PC, although they can be connected to pretty much any platform using the serial port.

6.3 Virtual Reality Hardware

- **Mattel Powerglove.** The Powerglove was originally manufactured for the Nintendo and although discontinued, is still widely available. It uses audio sensors to determine the position of the hand and the flexing of the finger joints. Interfaces are available for the IBM PC and the Macintosh computers. The cost for the glove is typically less than $50, and the interfaces range from public-domain circuits to an interface for the Apple Macintosh from TransInfinity (for $160). Abrams/Gentile Entertainment (AGE) also sell a serial interface that can be used on any platform with a serial port (for $200).

- **The Private Eye.** Reflection Technologies manufactures a head-mounted display device that can be used for virtual reality. The display is monoscopic and uses a spinning mirror to provide a resolution of 720x280. Cost is about $400.

6.4 Miscellaneous

- **Force Sensors.** Interlink sells force sensors called Force Sensing Resistors which transduce force into resistance. The force sensing pads are available for a few dollars for small ones, and about $50 for the large ones.
- **Stereo Sound.** Focal Point is a system for the Macintosh that can produce directional stereo (ie, binaural) sound. The complete package, with all the software and hardware needed costs about $1290.
- **Programmable Tactile Array.** The TiNi Alloy Company develops these for use in machine-human interfaces. Feedback is provided using mechanical simulation. A fully working demonstration package is available from Mondo-tronics for $149, and includes a single tactor (which can be mounted in a glove, for example), control box and software for the IBM PC. The interface is through a serial port.

7 Proposed Research

We would like to experiment with both hardware and software paradigms. In the best situation, we would develop a hardware user-interface using off-the-shelf components and experiment with various interaction methods for that specific interface. The quality of the interface would be judged using our criteria above.

7.1 Accessible Hardware

Access to the following three-dimensional hardware is available:

- **Mattel Powerglove.** The TransInfinity glove interface is used to connect it to a Macintosh. Demo software provided with the interface allows a user to move a virtual hand on the screen.

- **Logitech 3D mouse.** This is interfaced using the serial port. Demo software and drivers are available for the IBM PC and the Silicon Graphics workstations.

- **Citizen M329 LCD Monitors.** Two of these are available. Normal video and graphics can be output to these using the RasterOps graphics card for the Macintosh. These could be used to provide a stereoscopic display.

The total cost of the hardware is well under $3000.

7.2 Work in Progress

It was found that the resolution of the Powerglove was not sufficient to provide a good interface for fine interaction with a data set.

Therefore as part of this research effort the Logitech mouse was interfaced to an available Decstation and the software drivers were ported to it. Then some basic three-dimensional library routines were written. Finally, using the drivers and the three-dimensional library a rudimentary program was written that allows use of the mouse to position an object in space. The resolution was found to be sufficient, albeit a little low. However, the ease of positioning the dataset is unrivaled.

The next stage of work would involve writing a better interface, using some of the techniques that have already been found to be effective by other researchers. Since we also have access to a Silicon Graphics Indigo, it might be possible to use the GL library to provide the 3D graphics, which would provide much better performance.

References

[1] Grady, Sue. Here Comes the Visualization Workstation. *Computer Graphics Review.* Volume 4, No 3. March 1989.

[2] *IDL Reference Manual.* Research Systems Inc. Boulder, CO.

[3] *Dec AVS User's Guide.*

[4] Ware, C. and Jessome, D.R. Using the Bat: A Six-Dimensional Mouse for Object Placement. *IEEE Computer Graphics and Applications.* Volume 8, No 6. November 1988. pp. 65-70.

[5] Sachs, E., Roberts, A., Stoops, D. 3-Draw: A Tool for Designing 3D Shapes. *IEEE Computer Graphics and Applications.* Volume 11, No. 6. November 1991. pp. 18-26

[6] McWhorter, S., Hodges, L., Rodriguez, W. Evaluation of Display Parameters Affecting User Performance of an Interactive Task in a Virtual Environment. *Tech Report.* Graphics, Visualization and Usability Center. Georgia Institute of Technology.

[7] Eichenlaub, J. and Martens, A. 3D without glasses. *Information Display.* Vol. 8, No. 3, March 1992. pp. 9-12.

[8] Arielle Emmett.In Search of the Miracle Hologram. *Computer Graphics World.* Vol. 15, No. 7, February 1991. pp. 44-52.

[9] Kenneth and Debby Haines. Computer Graphics for Holography. *IEEE Computer Graphics and Applications.* Volume 12, No. 1. January 1992. pp. 37-46.

[10] Wall Street Journal item on holography.

[11] Bryson, S. and Levit, C. The Virtual Wind Tunnel. *IEEE Computer Graphics and Applications.* July 1992, pp. 25-34.

[12] Randy Pausch. Virtual Reality of Five Dollars a Day. *Proceedings of the ACM SIGCHI Human Factors in Computer Systems Conference.* April 1991.

[13] Stephanie Houde. Iterative Design of an Interface for Easy 3-D Direct Manipulation. *Proc. of the ACM SIGCHI Human Factors in Computer Systems Conference.* May 1992. pp. 135-142.

[14] Mercurio, P.J. and Erickson, T.D. Interactive Scientific Visualization: An Assessment of a Virtual Reality System. *Proc. of the IFIP, Third International Conference on Human-Computer Interfaces.* August 1990. pp. 741-745.

[15] Mackinlay, J.D., Card, S.K., Robertson, George G. Rapid Controlled Movement Through a Virtual 3D Workspace. *Computer Graphics.* Volume 24, No. 4, August 1990. pp. 171-176.

[16] Hiroo Iwata. Artificial Reality with Force-feedback: Development of Desktop Virtual Space with Compact Master Manipulator. *Computer Graphics.* Volume 24, No. 4. August 1990. pp. 165-170.

[17] S.Sydney Fels and Geoffrey E. Hinton. Building Adaptive Interfaces with Neural Networks: The Glove-Talk Pilot Study. *Proc. of the IFIP Third International Conference on Human-Computer Interaction.* August 1990, pp. 683-688.

8 APPENDIX A: Hardware Manufacturers

3DTV Corp.
P.o. Box Q
San Rafael, CA 94913-4316
Voice: (415) 479-3516
FAX: (415) 479-3316

Abrams/Gentile Entertainment, Inc.
244 West 54th Street, 9th Floor
New York, NY, 10019
Voice: (212) 757-0700

Dimension Technologies, Inc.
Rochester, New York.

Fake Space Labs
935 Hamilton Avenue
Menlo Park, CA 94025
Voice: (415) 688-1940
FAX: (415) 688-1949

Focal Point
318 W. Galer Street
Seattle, WA 98119
Voice: (206) 286-8922

Haitex Resources
P.O. Box20609
Charleston, SC 29413-0609
Voice: (803) 881-7518

Interlink Electronics
P.O. Box 40760
Santa Barbara, CA 93103
Voice: (805) 684-2100
FAX: (805) 684-8282

Mondo-Tronics
2476 Verna Court
San Leandro, CA 94577
Voice: (510) 351-5930
FAX: (510) 351-6955

Polhemus, Inc.
P.O. Box 560
1 Hercules Drive
Colchester, VT 05546
Voice: (802) 655-3159
FAX: (802) 655-1439

Reflection Technology
240 Bear Hill Road
Waltham, MA 02154
Voice: (617) 890-5905
FAX: (617) 890-5918

Sega America
3375 Arden Rd.
Hayward, CA 94545
Voice: (800) USA-SEGA

Spaceball Technologies
600 Suffolk Street
Lowell, MA 01854
Voice: (508)970-0330
FAX: (508)970-0199

TiNi Alloy Company
1621 Neptune Drive
San Leandro, CA 94577

Transfinite Systems Company, Inc.
Post Office Box N
MIT Branch Post Office
Cambridge, Massachusetts 02139-0903
Voice: (617) 969-9570

VPL Research Inc.
950 Tower Lane
14th Floor
Foster City, CA 94404
Voice: (415) 312-0200
FAX: (415) 361-1845 (?)

Appendix D:

Design and Development of a Data Visualization System in a Workstation Environment

Journal of Microcomputer Applications (1992) **15**, 81–88

Design and development of data visualization systems in a workstation environment

Gitta O. Domik and Kristina D. Mickus-Miceli

University of Colorado, Department of Computer Science, Boulder, CO. 80309-0430. USA and NASA Ames Research Center, MS T045-1, Moffett Field, CA 94035, USA

Data visualization has become an important component in the analysis of scientific data. This paper describes two objectives when extending data analysis environments by visualization: Offer visualization tools that fit the mind set of the targeted scientists; and integrate visualization tools into the complexity of existing data analysis tools. A case study performed between astrophysicists and computer scientists describes the design and development of STAR, a user-centered and integrated data analysis and visualization system.

1. Introduction: the role of data visualization

During the last five years, scientific visualization has found a multitude of new applications. With an increase of computing power seen in low-end, mid-range and supercomputing systems, as well as increased data rates from data-collecting sensors (higher resolutions, higher transmittal rates, compact storage devices), the availability of scientific data has multiplied. The increased need to interpret large amounts of data and the easy-to-understand principles (encode numbers into pictures) led to a recent popularity of scientific visualization. Encoding numbers as pictures stimulates mental processes in a different way than the interpretation of numerical values: Pictures allow the observer to browse in a large data set to examine large-scale structures and to focus at any time to observe small-scale structures; pictures emphasize spatial relationships between objects described in a data set; and pictures may suggest patterns not obvious when looking at the representative numbers. Such qualitative judgments are made possible through data visualization, whereas numerical values lead to quantitative judgments of the data to be interpreted.

Over the past five years, much progress has been made on the use of computer graphics to represent complex data structures visually: e.g. volumetric data, multidimensional data, or fluids. The advent of scientific visualization influences various disciplines, but none as greatly as computer graphics, where the demand for simulations of complex physical events and complex data structures is stimulating and challenging to the experts in this field. While complex computer graphics techniques are essential to any progress in scientific visualization, there are other issues at hand that need the attention of experts in various fields, such as software engineering, user interface design, perception, artificial intelligence, cognitive sciences and related fields, if comprehensive solutions to increase scientific productivity are to be built.

In spite of many examples of applications in scientific visualization, an agreement on concepts and models, qualitative and quantitative measures of the 'usefulness' of

0745–7138/92/020081 + 08 $03.00/0

visualization, and a solid basis of its principles in human perception are still missing. Crucial questions such as 'What are our limits in merging information simultaneously?', or 'How does colour affect our ability to interpret data?' or 'What is the role of interactivity in exploring complex data sets?' cannot yet be answered in a satisfactory way [10]. Many of the new visualization techniques are not yet in the hands of the end-user, but still in 'Visualization Labs' with a high ratio of computer scientists to scientists.

It is the latter problem that this paper addresses. In a collaboration between astrophysicists (called 'scientists' throughout this paper) and computer scientists (called 'designers' throughout this paper), we extended an existing data-analysis environment with visualization tools. Our focus is on building user-centered visualization tools and aiming towards an integrated scientific visualization software.

2. The environment of the case study

In early 1990, scientists at the Center for Astrophysics and Space Astronomy (CASA) at the University of Colorado, decided to expand CASA's data analysis environment to take better advantage of large amounts of scientific data available to the astrophysical community [7,4,2]. The data are described by heterogeneous data structures (zero, one, two and multi dimensional data) and diverse data characteristics, such as varying spatial and spectral resolutions. An improvement in data visualization and interaction techniques held the promise for better exploitation of available data. It is necessary to first understand the working environment of the scientists to be aware of the complex and distributed software environment available at CASA, typically for a scientific environment based on data analysis.

CASA hosts fifteen scientists and about the same number of graduate students. The bond between their various research interests is established by large amounts of data acquired from space and ground-based observations and a network of hard- and software to manipulate such data. Images, spectra, tables and text compose the greater portion of CASA's data bases. Scientists require access to all available data bases to ensure that a complete inventory of information is at their disposal to pursue their research. Following identification and retrieval of data, preprocessing is often necessary to deal with noisy data and to remove and correct instrumental effects. Subsequent numerical calculations, often in the form of statistical analysis, together with visual and interactive data processing, constitute the most significant modules of a scientific data analysis software system [7].

Most of the software used for identification, retrieval and preprocessing of data items has been developed by CASA's scientists, students and staff. Public domain software covers most of the numerical and visual analysis modules. Due to the various characteristics of space and ground sensor data from different wavelength ranges (such as radio, infrared, visible or x-ray), different software packages are being used. Input and output data streams are not fully standardized, reflecting the lack of standard data formats, and resulting in 'islands' of software systems [9] as depicted in Fig. 1. Each software package has its strengths and weaknesses in its ability to preprocess, analyse and visualize astronomical data of specific characteristics. Integration of software systems into a common user environment to perform multi-spectral and multi-sensor data analysis and visualization demands sharing of data [3,8,12].

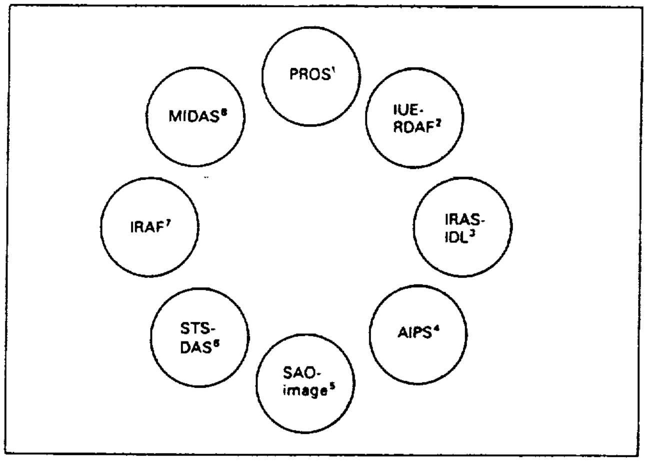


Figure 1. A subset of public domain software packages used for astronomical data analysis.
1. PROS: Post-Reduction Off-line Software; 2. IUE-RDAF: International Ultraviolet Explorer Regional Data Analysis Facility (IDL library with IUE specific functions); 3. IRAS-IDL: IDL library with IRAS specific functions; 4. AIPS: Astronomical Image Processing System; 5. SAOimage: Image display software developed by the Smithonian Astrophysical Observatory; 6. STSDAS: Space Telescope Science Data Analysis System; 7. IRAF: Image Processing and Analysis Facility; 8. MIDAS: Munich Image Data Analysis System.

The computing environment consists of graphics workstations of type VAXstations, DECstations, and SUN/Sparcstations. The diverse representation of floating point data in these three architectures additionally complicates travelling between various software packages and data formats.

CASA's scientists have long appreciated the knowledge gained by utilizing multisensor and multi-wavelengths data, but have been hampered by the complex software environment created by the simultaneous use of various software packages and the need to convert various data formats. The plan to develop multi sensor data sets demanded an integration of existing tools and data.

3. Collaborative design

User needs and resources need to be balanced for a comprehensive and realistic visualization solution. By generalizing the user's demands, the need for resources increases, and the visualization representations might not fit the mind set of the specific user. Therefore the design and development of visualization software was to centre on the specific users at CASA, scientists with a variety of needs to serve their various research interests but of similar education and doctrines. The 'user-model' to drive the new visualization software named STAR (Scientific Toolkit for Astrophysical Research) was to be defined by computer scientists and astrophysicists in a collaborative effort. Real solutions to active problems preceded graphically challenging visualizations.

In order to solicit the scientists' opinions on their desires for user interfaces and visualization tools, cognitive methods were used. These methods could be shown in the past to improve the quality of interaction between designers and users, leading to active user participation in the design process and to user-centered solutions [6].

As a means to initiate discussions between collaborators, a research scenario of limited scope was defined by designers and scientists. It reflected the main tasks of a data analysis and visualization system to solve a specific problem, namely the analysis and display of multi-spectral data: Infrared data of various wavelengths responses, but of a specific spatial location, were searched for in CASA's local data bases; resulting images were visually checked for calibration errors and were preprocessed, if necessary; statistical analysis in form of numerical calculations (e.g. flux integration over an extended area) or graphical representations (e.g. vertical/horizontal profiles in the area of interest) were performed next; finally, three representative data sets were merged into one colour image using red-green-blue or hue-lightness-saturation colour coding. An initial prototype of STAR was built around a direct manipulation (point-and-click) user interface to reflect the goals of an integrated and user-centered visualization solution and tested against the specific research scenario. With this initial testbed, individual user interviews and general systems demonstrations could start.

At individual user interviews, scientists were given the tasks of the research scenario to perform in a 'Thinking Aloud' session [5], or they commented on the user interface of STAR and its functions in a one-to-one interview with a designer. The 'Thinking Aloud' method consists of a user trying to perform the tasks outlined by the designer, with the user continuously giving comments about his/her success. Interaction between designers/users is not encouraged during the Thinking Aloud session. The comments are taped and later transcribed and discussed by designers/users. One-to-one interviews between designers and users create an interactive discussion, with questions/answers. Individual user interviews resulted in a multitude of changes in the design of STAR, meeting our goal of an iterative design.

General demonstrations promoted the existence of STAR at CASA and decreased the general concern about building 'another software package that will not interface with the current environment'. Besides talks and on-line demonstrations, we also provided an easily accessible poster board to describe STAR's user interface and functionality; in cognitive terms this is called a 'storyboard technique'. Individual user interviews were obviously much richer and useful in their feedback; however, general demonstrations as well as the poster promoted the system to outsiders and initiated talks and discussions.

One of the major problems in a collaborative design with scientists is to get input from scientists, as their time is usually already oversubscribed. A paper [7] describes in more detail issues surrounding a collaborative design to build scientific software systems.

4. Integration of visualization into a complex software environment

Expanding the analysis cycle through visualization and interactivity is necessary to prepare current data-analysis software for use with large sets of diverse scientific data. As discussed in section 2, many different tasks need to be performed between the acquisition of scientific data and its interpretation by a scientist. These tasks are often represented by different software packages. The scientist trying to solve an individual scientific question is challenged by a complex conglomerate of software systems distributed on various workstations and is thus hampered in his/her scientific work. The development of visualization tools only adds to this complexity.

In order to reduce the complexity as experienced by the user, we first unified existing software packages and tools under a common user interface and then integrated any new tools into the new software environment. The development platform at CASA for in-house developments by scientists and staff is IDL*. As STAR was also developed in IDL, integration with in-house developed software could be accomplished with little effort. IDL software packages are collected in source libraries. Depending on the type of data to be analysed, the user would click one of several round knobs on the user interface indicating the correct data type and thus identify the libraries to activate (Fig. 3). Activating all libraries at the same time will reduce the speed of the execution of individual modules as well as exhibit conflicts between modules of the same name with different functions. The user may switch from one IDL library to another at his/her convenience. Each software package is organized according to the data analysis loop Domik 1991b: access, identify, retrieve data; pre-processing; analysis; visualization.

Software packages developed in other languages are not as easy to integrate. Square buttons on the user interface indicate their existence and allow access to such 'foreign' packages. Travelling between different packages demands a common solution to the various data formats. The astronomical community has spent much effort in designing the common data format FITS† over the years, resulting in several extensions. A suitable FITS definition needed to be chosen as STAR's common data format. Data conversion (to/from FITS format) is handled by the main menu 'Data I/O'‡. The user may switch between various foreign analysis packages, entering a new user interface when doing so, and returning to STAR's main user interface after exiting such a software session. An additional menu 'Environment Options' contains modules for administrative tasks.

The solution as described above is shared by diverse workstations, as IDL can be implemented on a variety of platforms. This was an additional advantage for choosing IDL as our prototyping/developing language, as the workstation environment at CASA is constantly expanded and new hardware acquired.

5. Reflections on user-centered and integrated solutions

The user interface to our initial prototype, before user interviews and demonstrations started out, is reflected in Fig. 2. To date it has changed to Fig. 3. Many functions reflected by menus and buttons in Fig. 3 were not available at that time; individual menus have been extended. As a result to the iterative, collaborative design, the following changes have been implemented in STAR:

- A status window to inform the user about progress of issued commands or the general condition of the system.

* Interactive Data Language by Research Systems, Inc.

† Format Interchange Transfer System.

‡ Many of the data conversion programs were made available by the IDL Astronomy Library.

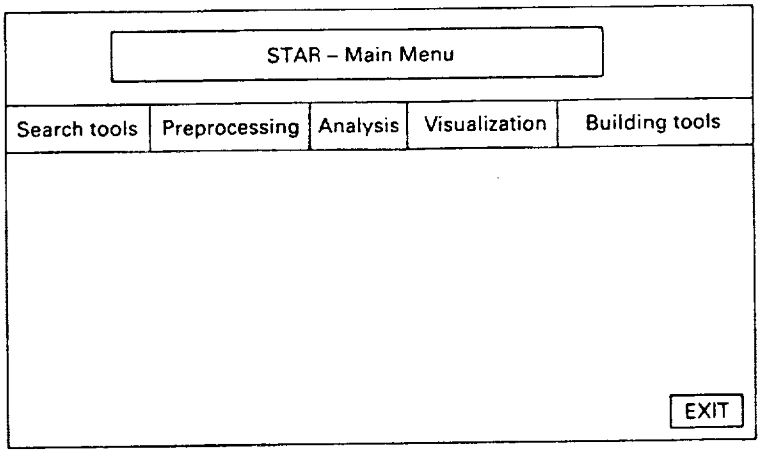


Figure 2. STAR's original user interface.

- Problems often occur when the designers/developers are not available. Frustration about an error often passes until the next encounter between scientist and designer. Therefore a problem button was implemented to vent the users' frustration and document the problem on the spot. The mailing system sends an appropriate message immediately to the designers. The location and size of the problem button experienced many redesigns.

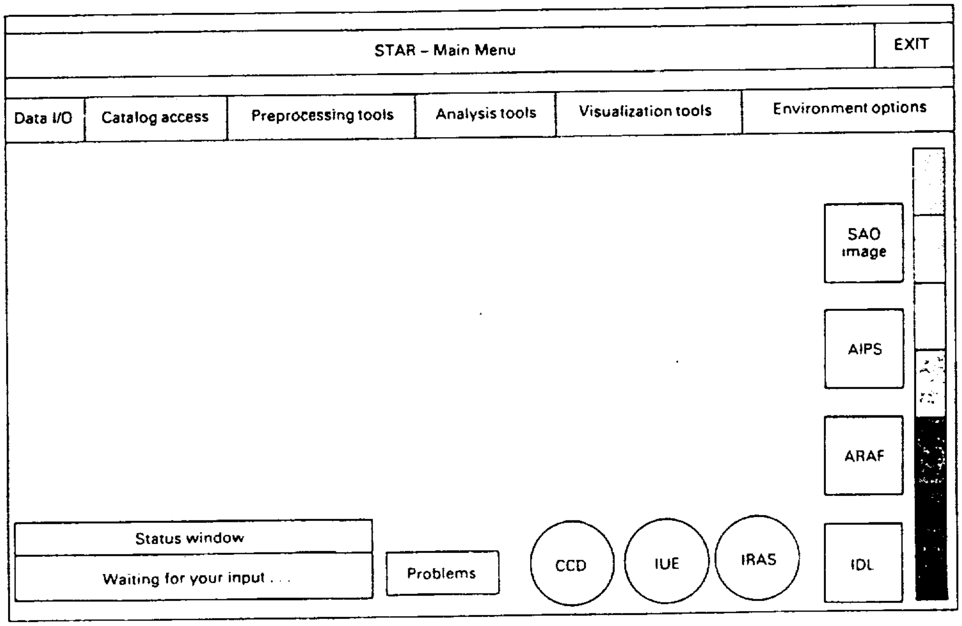


Figure 3. STAR's toplevel user interface.

- The 'exit button' in the first prototype could not be located by many users; as with the 'problem button', its location and size was the issue of many discussions and redesigns.
- The visual representation of quantitative information demands a well understood mapping of colour/brightness to quantity. Therefore a scale bar was demanded as part of the permanent user interface layout.

A surprise to the designers was the demand of visualization tools: interactive tools to analyse and display data (e.g. interactive profiling, interactive manipulation of lookup tables; or interactive mapping of data values from a high dynamic range to 256 available colours) were much more in demand than complex visualization tools (e.g. transparency, or data slicers).

The current installation of STAR runs on all graphics workstations at CASA. Not all foreign packages are compatible with the wide range of IDL installations, so that the user interface will not look the same on every hardware.

The current status of STAR reflects two years of work in participatory design and development of a visualization system integrated into a complex data analysis environment. Next steps in the still ongoing project are further expansions of STAR's integrated software environment and the addition of more complex visualization tools.

Acknowledgments

The authors are grateful to all the scientists at the Center for Astrophysics and Space Astronomy, specifically to Dr Edward W. Brugel, Dr Thomas R. Ayres, Dr Bob Stencel and all the graduate students involved, for their collaboration. Scott Merkle and Anthony Veale greatly supported the development of STAR. A more elaborate discussion on the topics described in this paper, specifically on participatory interface design, can be found in Kristina Mickus-Miceli's thesis on 'Participatory User Interface Design for Scientific Visualization Systems'. Reflections and conclusions expressed in this paper are based on her efforts during her work towards her Master's degree at the University of Colorado. Prof. Clayton Lewis gave very helpful advice throughout the duration of this work. The project is supported by NASA under grant NAGW-1902. Gitta Domik's efforts were partly also sponsored under NSF grant ASC-9001115.

References

1. G. O. Domik, 1991*a*, The Role of Visualization in Understanding Data, Lecture notes on Computer Science 555, Springer-Verlag, *'New Trends and Results in Computer Science'*, 91–107.
2. G. O. Domik, 1991*b*, Visualization Techniques to Aid in the Analysis of Multi-Spectral Astrophysical Data Sets, *Final Report on NASA Project NAGW–1902.*
3. G. O. Domik, 1991*c*, Software Design and Development in a Scientific Environment: Lessons Learned During the Development of STAR, an Astrophysical Analysis and Visualization Package, *First Annual Conference on Astronomical Data Analysis Software and Systems, November* 6–8, 1991.
4. G. Domik & K. D. Mickus, 1991, Visualization in the Analysis Cycle of Observational Data, *Proceeding of Gesellschaft für Klassifikation*, Salzburg, February 25–27, 1991.
5. C. Lewis, 1982, Using the 'Thinking Aloud' Method on Cognitive Interface Design. *Technical report.* IBM Thomas Watson Research Center, Yorktown Heights, NY, 10598, 1982.

6. C. Lewis, 1989, A Research Agenda for the Nineties in Human-Computer Interaction. *Paper representing a talk given at the Winter Conference of Human-Computer Interaction Consortium, Vail, Colorado*, January 22–24, 1989.
7. K. D. Mickus-Miceli & G. O. Domik, 1991, Participatory Design in the development of a Scientific Software System, *Technical report*, Department of Computer Science, University of Colorado, Boulder, CO, 80309-0430.
8. S. Murray & E. Mandel, 1991, The ASSIST User Interface Prototype, *First Annual Conference on Astronomical Data Analysis Software and Systems*, November 6–8, 1991.
9. D. R. Nadeau, T. T. Elvins & M. J. Bailey, 1991, Image Handling in a Multi-vendor Environment, *Proceedings Visualization '91*, G. M. Nielson & I. Rosenblum, eds), IEEE Computer Society Press.
10. G. M. Nielson, 1991, Visualization in Scientific and Engineering Computing, *IEEE Computer*, **24**, pp. 58–66.
11. Prawel. D., 1991, How Visualization Applications Drive Tool Selection: One Product Can't Do It All, *Proceedings Visualization '91*, G. M. Nielson & L. Rosenblum, eds. IEEE Computer Society Press.

Gitta O. Domik is Assistant Research Professor at the Department of Computer Science at the University of Colorado at Boulder. Her research interests include scientific visualization, image processing and computer graphics, with a significant side interest in user interfaces, scientific data bases and knowledge based systems. On her work in scientific data analysis she collaborates closely with the scientists, therefore building a bridge between the needs of scientific users and new methodologies in computer science. She teaches courses in digital image processing and scientific visualization as well as 3-D computer graphics. Gitta graduated with a PhD in Technical Sciences from the Graz University of Technology in Austria in 1985. She is a member of IEEE and ACM.

Kristina D. Mickus-Miceli is a PhD student in Computer Science at the University of Colorado, Boulder. She is working on her dissertation under fellowship at NASA Ames Research Center on the topic of data models for multidisciplinary visualization. Mickus-Miceli received her BS degree in Aerospace Engineering in 1989 from the University of Colorado. She received her MS in Computer Science in 1991 at Colorado, where she completed her master's thesis on participatory user interface design for scientific visualization systems.