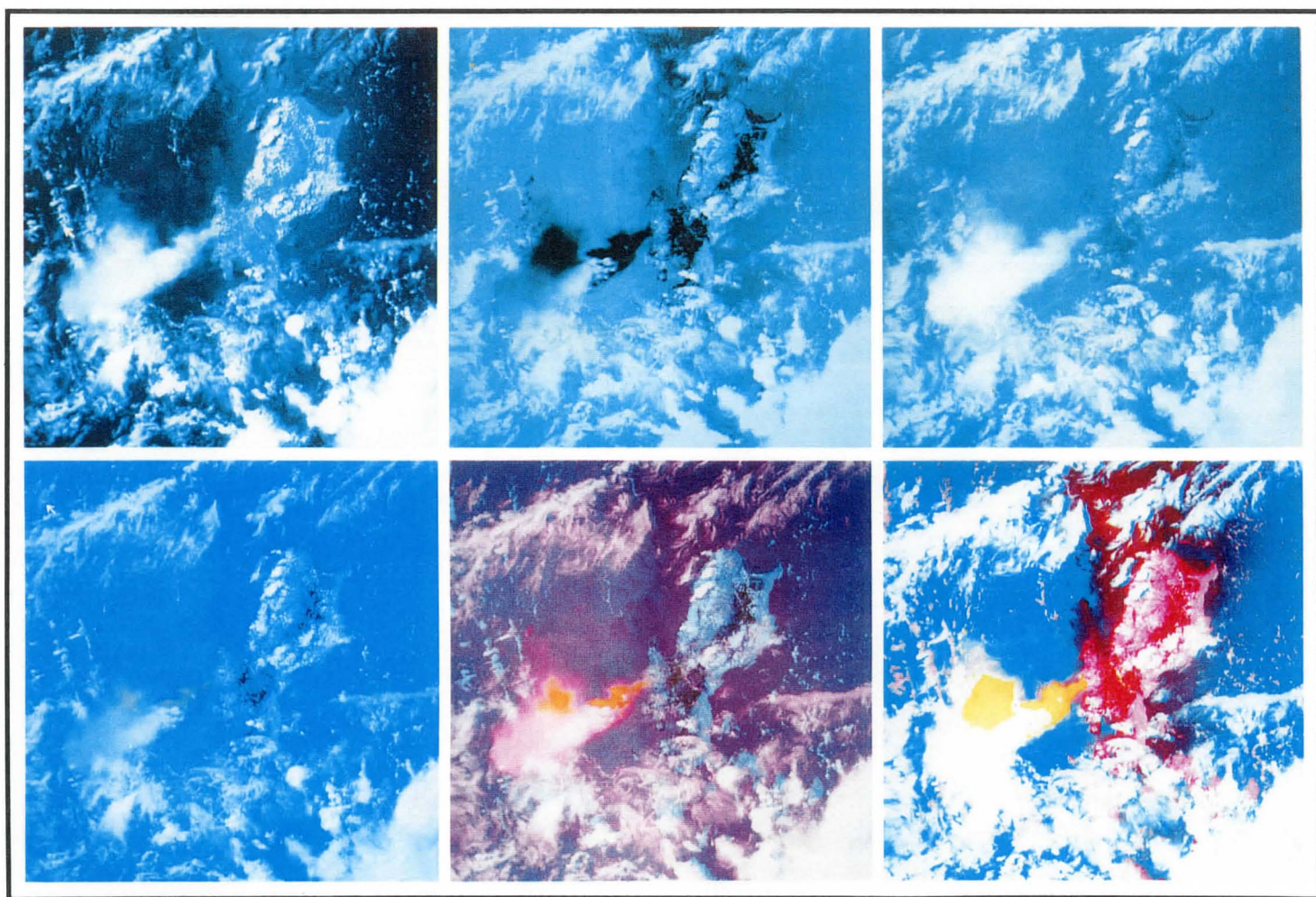


NINTH INTERNATIONAL CONFERENCE ON INTERACTIVE INFORMATION AND PROCESSING SYSTEMS FOR METEOROLOGY, OCEANOGRAPHY, AND HYDROLOGY

January 17 - 22, 1993

Anaheim, California



AMERICAN METEOROLOGICAL SOCIETY

EXPLORATORY ANALYSIS OF SATELLITE DATA USING THE INTERACTIVE IMAGE SPREADSHEET (IISS) ENVIRONMENT

K. Palaniappan, A. F. Hasler & M. Manyin

Laboratory for Atmospheres
NASA/Goddard Space Flight Center
Greenbelt, MD 20771

1. INTRODUCTION

The traditional numerical spreadsheet paradigm has been extended to create an Interactive Image SpreadSheet (IISS) Environment. The concept of the IISS is described by Hasler *et al.* (1991a, 1992) and has been developed to interactively examine and manipulate large remote sensing data sets. The IISS Environment is currently being used to analyze multi-spectral data sets such as complete GOES/VISSR, Landsat TM, AVIRIS and NOAA/AVHRR imagery in anticipation of data from EOS satellite instruments. The IISS Environment provides a spreadsheet-based visual interface for performing satellite image analysis tasks. Such an interface has been found to be extremely intuitive and highly productive since it eliminates to a large extent the necessity for a user to explicitly deal with input/output based programming as with traditional computer languages. The resulting IISS Environment provides the scientist with an effective and powerful visualization tool for concentrating on algorithm development and exploring the information available in the data.

The IISS Environment, as with traditional numerical spreadsheets, enables the interactive organization, manipulation and processing of large amounts of data in an intuitively comprehensible manner. The IISS Environment, however, is inherently three-dimensional to accommodate organizing data in the time or channel dimension. Each page or layer of the spreadsheet can manipulate an independent group of data with the added flexibility that relationships between spreadsheet cells and layers can be directly specified. Each cell of the image spreadsheet contains one or more multidimensional data sets to be visualized. Visualization data sets include raw and processed satellite imagery, graphical (vector) data, surface and terrain models, and three-dimensional volumes.

A powerful feature of spreadsheets is their formula evaluation capabilities. Using a similar syntax for defining formulas that operate on images, vectors, surfaces and volumes provides a powerful exploratory analysis tool in the IISS Environment. Common mathematical and visualization operations are immediately accessible in the IISS Environment; these include band ratioing, linear combinations of bands, nonlinear algebraic operations on pixels, logic-based thresholding, contrast enhancements, false color-composites, animation of time series imagery, combining visual imagery with an elevation height field for surface rendering and stereoscopic viewing, and isosurfaces of visual cubes. Such a diverse set of visualization tools in combination with an extensive library of mathematical image operators provides a productive exploratory analysis environment for working with diverse satellite data sets.

2. THE IISS ENVIRONMENT

A typical image spreadsheet may range from a 2 X 2 to an 8 X 8 matrix of cells where each cell contains an image. The data structure used to support interaction with the image

spreadsheet and facilitate image operations is described in Section 2.1. Some of the features of the graphical user interface and dialog forms are described in Section 2.2, and Section 2.3 summarizes the major visualization modules. The analytical tools provided by the IISS environment using formula evaluation are described in Section 2.4.

2.1 Data Structure Description

A hierarchical tree-based data structure was designed to represent the image spreadsheet and facilitate the interactive manipulation of the image data residing in cells of the spreadsheet. A hierarchical scheme was chosen to reflect the actual organization of the image spreadsheet and to define objects that closely corresponded to items being manipulated by the user. The terms Sheet, Cell, Frame and Image are shown illustrated in Fig. 1. The image data is organized in the sheet in terms of frames that are grouped together in cells. Within each cell is a set of frames termed a Framestack that can be considered to be in the depth dimension of the image spreadsheet. Each layer of the data structure is governed by a specific set of functions which hide implementation details from the layer above. This object oriented approach both simplifies software development, and prevents functions that manipulate small parts of the data structure from unintentionally changing larger portions of the structure.

The chart in Fig. 2 shows the overall hierarchical organization of the IISS data structure. The image spreadsheet data structure contains all of the data required to define, display and manipulate a set of image data. Members of the data structure at the same level in the hierarchy have the same parent data field. The Sheet is a two-dimensional array of Cells with each cell initially being of the same screen dimensions. Cells are arranged in a matrix form and are accessed via matrix addressing since the two-dimensional arrangement of cells will typically remain stable during a user's session or will change infrequently. Resizing an individual cell is disallowed, because it would upset the regular organization of the spreadsheet that affords its ease of interpretability. However, rows and columns will be resizable, since the matrix appearance is preserved under these operations. There are no a priori limitations on the size of the matrix of cells that can be created and manipulated; the image spreadsheet's size (the number of rows and columns of cells) can also be dynamically changed by the user. Additional information at the Sheet level includes screen size of the sheet, titles of rows and columns, screen size of the cells, and hardware features and limitations like double buffering the screen display, the size of the colormap (8 bits versus 12 bits for example), or the capability to zoom an image by fractional amounts. Note that some leaves in the tree represent individual members of the data structure while others refer to categories of items coalesced to simplify the chart. For example, the Geometric Transformation field would include the scaling, translation and rotation parameters for displaying an image.

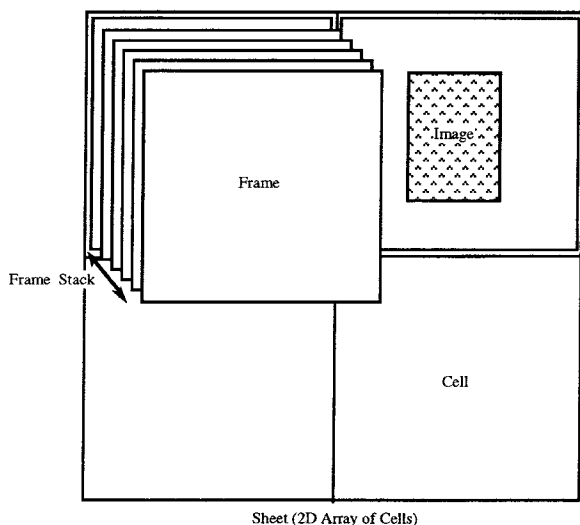


Figure 1. A diagrammatic representation of the IISS cell data structure. The illustration shows the terminology for the composition of the sheet.

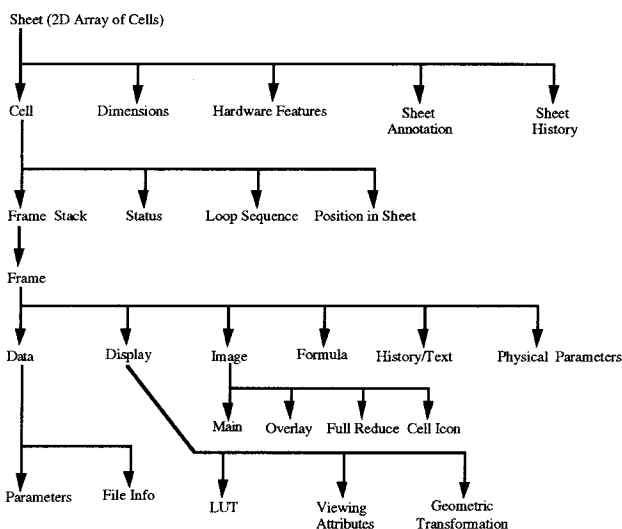


Figure 2. IISS hierarchical object oriented data structure showing the organization of the image spreadsheet.

Each cell contains one or more frames arranged as a linked list and denoted by the term Frame Stack. Two groups of pointers are maintained, one to keep track of all the frames in the Cell and another to keep track of the animation sequence when a subset of frames are looped together. Cells that are grouped together can be accessed by a linked list of pointers to cells that are part of the group.

The frame data structure is the most complex since it contains all of the information pertaining to the image that is visible on the display as well as the original source of the image data.

2.2 Graphical User Interface

An effective user interface should facilitate ease of use, consistency, portability, extensibility and maintainability. In order to provide a consistent look and feel, and reduce development and maintenance time, the use of a pre-existing library of user interface "building blocks" is considered essential to aid in program development. The **forms** library written by Mark H. Overmars of Utrecht University in the Netherlands and

based on the Silicon Graphics Computer Systems Graphics Library (**GL**) was chosen for several reasons. The **forms** library provides high-level functionality, allowing the programmer to accomplish with one function call the equivalent of many Motif function calls. It is relatively easy to learn, produces compact executable code as compared with other libraries, and its event handling is versatile, allowing for both callbacks and event queue processing. The source code for **forms** is available thus allowing custom modification.

On the issue of portability, **forms** depends on **GL**, which currently requires dedicated hardware or proprietary SGI software. Fortunately, **GL** has been licensed to other manufacturers and the family of machines which support **GL** is steadily growing and now includes all SGI machines, the IBM RS/6000 series, as well as some DEC and SUN SPARC workstations. The next generation of **GL**, "OpenGL", is expected to be supported by many additional vendors, including Microsoft, Intel, Compaq and AT&T/Unix Systems Labs. Provisions are in place to ensure the smooth migration of the IISS to use OpenGL when **forms** makes the transition.

User interaction with cells includes both popup menus for accessing module commands, and keystroke accelerators for a subset of these. The user is given control of a cell simply by moving the mouse cursor to within its borders. This unobtrusive interface maximizes viewing area at all times. The interface forms need not be displayed on the same screen as the main Spreadsheet window. The current version of **GL** supports the display of graphics across a network on workstations supporting Distributed **GL**. So a low cost Silicon Graphics Personal IRIS, or IRIS Indigo can be used to display the graphical user interface.

Examples of several IISS user interface forms including the control panel and the formula dialogue form are illustrated in Section 4. The form that appears at the left of the figures is a control panel that provides scrollbar navigation over the cell rows and columns, data input/output selectors, module activation, and feedback fields.

Dataflow or block diagram based visual interfaces to applications use nodes to represent processing modules and interconnections between nodes to indicate the movement of data from node to node. The dataflow approach to describing and executing algorithms has been used previously for image processing applications (Fisher, 1988). The Cantata tool in Khoros is a more complete implementation of a visual programming environment for signal and image processing incorporating conditional and iteration constructs, shared memory data transport between nodes, and distributed processing of nodes across remote machines (Rasure and Williams, 1991). The Khoros library of image processing functions is used to implement some of the standard operations in the IISS Environment. Visual language models can be used to help the user keep track of the complex processing steps needed to analyze large multichannel data sets. As the features supported by the IISS evolve, the complexity of formulas and operations between cells and frames may necessitate a visual data flow interface to keep track of and modify frame formulas.

2.3 Data Visualization Modules

An inherent feature of the IISS is the capability of very quickly scanning or browsing through immense data sets. This IISS *super browse* capability allows the rapid scanning in addition to careful examination of data in the spatial, temporal, spectral, and gray scale domains. The IISS can display data in 2-D, 2.5-D and full 3-D rendered modes as well as full color binocular stereo with animation of time series for all of the above. *Super browsing* of huge data sets is made possible by the software modules described here and the high performance of the workstation and mass storage system described below in Section 3.

The *spatial browser* uses the **rgb_zoom** module for high performance interactive roam, zoom, and reduce operations on 24-bit arbitrary sized images. The image data can occupy up to 230 MB of RAM without sacrificing performance using the current hardware configuration described in Section 3. Synchronized smooth and fast zoom and roam operations have been demonstrated using three Landsat TM scenes each 8940 X 8420 pixels that require at least 226 MB of RAM. Each scene occupies one cell of a 2 X 2 spreadsheet and scale and position changes in one cell are immediately reflected in the other three cells. Another large data set for which interactive image rescaling and image translation has been found to be very useful, are from the GOES satellite VISSR instrument that produces visible images every half-hour at 15200 X 12000 pixel spatial resolution with six bits per pixel brightness quantization (that is typically stored and displayed at eight bits or one byte per pixel). A single GOES/VISSR image requires 182 MB of memory just for the visible channel. All of the frames within a cell are also optionally updated to reflect the current frame parameters; this is useful for comparing features in large multi-band images of the same scene.

The *temporal* and *spectral browsers* use features of the **loop** module that allows effective viewing of image sequences in groups of cells. Image sequences may be long time series data (i.e. 900+ frames) or bulky multichannel data (i.e. 224 channels). Supported features include: (i) asynchronous frame animation, (ii) spatial synchronous frame animation, (iii) inter-cell frame animation, (iv) multichannel synchronized frame animation, (v) time selected frame animation, and (vi) stereo frame animation.

The *dynamic range browser* uses the **lut** module, which is currently being designed, to examine data sets which have more than the standard 8 bits (256 levels) of gray scale information. Many sensors like the AVHRR instrument provide 10 bit measurements. The **lut** module will provide interactive capability to adjust the dynamic range and contrast of the data. New methods for interactively displaying data sets with more than 12 bits (> 4096 levels) per measurement are also being developed.

The *volume browser* uses the **flyby** module (developed at Goddard) and may also use features of the **vis-5d** module developed at the Univ. of Wisconsin (Hibbard and Santek, 1991) to browse through 3-D and 4-D observed and simulated (numerical model) data sets. Stereo imagery of 3-D data can be animated using the **im_stereo** module for binocular stereo presentation of 24-bit image pairs using the Stereographics hardware. The IISS volume browser is still being designed and will also provide features to develop data fly-through animations as is currently available in the standalone **flyby** module. The **flyby** module is used to display and manipulate surface data which typically require one to four data fields (i.e. height field alone or three data fields to define a 24 bit textured image that is mapped on a height field). The **flyby** module can also be extended to visualize data sets mapped on a globe as described in Hon (1991).

2.4 Analytical Tools

Each cell and frame in the image spreadsheet supports the composition and analysis of formulas using operators acting upon data sets contained in one or more frames. Standard mathematical functions, such as addition, subtraction, multiplication, division, mean, standard deviation etc., may be performed easily on individual image cells or on groups of two or more images. The IISS formula language is inherently an image processing language. Some of the features of the formula language are based upon IAX developed at IBM UK Laboratories (Jackson, 1988) and the syntax used in the symbolic mathematics package Mathematica (Wolfram, 1991) or current spreadsheets such as Lotus 1-2-3 or Microsoft Excel.

The formula notation consists of methods for addressing elements such as cells or frames, defining ranges of indices to access more than one element conveniently, and a set of supported operators and functions. Rows of the spreadsheet are referenced by letters and columns by numbers starting with cell A1. Frames in a cell are numbered from one. Rows and columns of a single frame are indexed from (0,0) representing the upper left corner. For example, A1[[2, 12, 100]] addresses pixel element in column 100 of row 12 in frame 2 of Cell A1. Ranges can be specified explicitly using operators such as Row[], Column[], Frame[] or Cell[] or implicitly using colon or double dot notation and wild card symbols.

Many operators are overloaded to accommodate indexing over all pixels in a frame or image. For example, a simple addition expression such as

$$B1 = A1 + A2 \quad (2.1)$$

actually implies summation over all corresponding pixel elements in corresponding frames in cell A1 and A2 with the result being placed into corresponding frames in cell B1. Note that the frames are also implied in this notation. In Mathematica-like notation the expression above would be:

$$B1 = \{A1[[f,x,y]] + A2[[f,x,y]], \{x,0,255\}, \{y,0,255\}\} \quad (2.2)$$

Expressions need not explicitly identify the frames on which the operations are to be applied, as in the example above. Expressions that are based on the *current frame* (currently viewed frame) rather than an absolute frame index, need to be resolved carefully since the current frame index may be altered at any time.

The dynamic update or data-driven computational model is usually the default mode of operation in standard spreadsheets since the calculations are usually rapid. However, in the IISS environment operations can be time consuming due to the volume of data and complexity of algorithms, so the default mode of operation would be *evaluate upon user request* which is the demand driven model of execution. Individual cells can be in auto update (data-driven) or manual update (demand-driven) modes depending upon the degree of real-time responsiveness required by the user.

Arithmetic, logical and bit operators would be supported uniformly for scalars, vectors and 2-D matrices (images). A small set of mathematical and numerical functions such as absolute value, trigonometric functions and convolution operators would be supported for scalars, vectors and matrices appropriately. Functions that are more specific to images and involve image analysis algorithms include flipping data values about various axis, local statistics calculations, thresholding, filtering and histogram operations. Typical arguments to these functions would include specification of one or more frames, and definition of parameters that the function requires. The IISS formula expressions that involve image processing operations are parsed and converted to a series of Khoros library calls (Rasure and Williams, 1991). The advantage of using the Khoros library for implementing image processing functions is that it is readily available and distributed free of charge by the Khoros Group at the Univ. of New Mexico, portable across personal computers to supercomputers, and widely used with an active library of user contributed modules for complex image analysis algorithms.

Formula expressions also support image spreadsheet functions for conveniently copying or duplicating information between cells and frames, for adjusting rows and columns, and accessing groups of cells. An Element[] operator is used to retrieve the displayed data values.

A complex example could be done in one step or each step could be distributed to different frames. The following algorithm, described in Section 4.2, is used to detect thin cirrus clouds. There are two threshold operations as well as a multiplication, division and subtraction on a corresponding

INTERACTIVE IMAGE SPREADSHEET (IISS) WORKSTATION & SERVERS

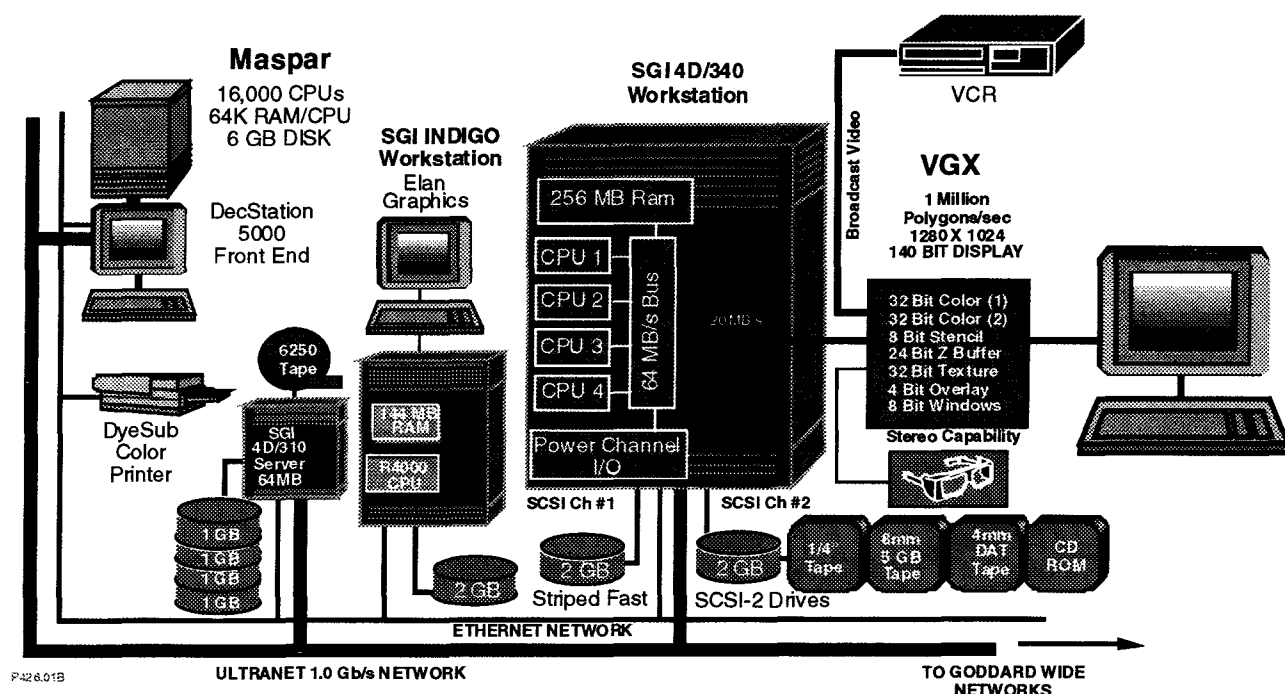


Figure 3. IISS hardware environment showing the high performance SGI 4D/340 VGX and the Indigo Elan workstations.

pixel-by-pixel basis throughout the entire image in frames A1[[1]] and A2[[1]] with the result being written to Cell B2, Frame 1.

$$B2[[1]] = \text{Threshold}[\begin{matrix} 100 * A2[[1]] / \\ (A1[[1]] - \text{Threshold}[A1[[1]], 100, 255]), \\ 75, 200 \end{matrix}] \quad (2.3)$$

3. IISS HARDWARE ENVIRONMENT

A Silicon Graphics Computer Systems (SGI) supercomputing workstation SGI 4D/340 VGX, as illustrated in Fig. 3, is being used as the primary platform to demonstrate the IISS. The 4D/340 VGX has four MIPS R3000 CPUs with benchmarks of 120 MIPS and 36 MFlops, random access memory (RAM) of 256 MB, high performance (20 MB/s) data transfer between RAM and display frame buffer. The VGX graphics offers exceptional performance at one million triangles per second. The high resolution monitor (1280 X 1024) has 140 bit planes available for graphics operations. A distributed mass storage system using multiple disk drives totaling 10 GB is available with 2 to 5 MB/s access rates for locally mounted hard drives. Parallelizing C and Fortran compilers have been used to take moderate advantage of the multiprocessing and multitasking capabilities of the 4D/340 workstation.

The 256 MB of RAM is used to display large IISS arrays interactively and ensure that long animations (900 frame 8 bit and 300 frame 24 bit color 512 X 512 pixel) can be accessed at a minimum of 30 frames/sec. New fast SCSI-2 hard disk performance access to the local mass storage system at a rate of 3.3 MB/s has been demonstrated. Striping of two disks will support transfer rates of 5-6 MB/s. The high performance (125 MB/sec) UltraNet network interconnect has been installed to access remote storage. UltraNet is used to access the SGI 4D/310 Server, and the Maspar large scale parallel processing computer shown in

Fig. 3 as well as Goddard databases and other super computers. The Maspar has been used extensively for image processing tasks such as automatic stereo analysis (Hasler, 1991b).

The four processor SGI 4D/340 VGX workstation allows production model runs, software development, software testing and demonstration to proceed concurrently with a minimum of interference. Some of the 3D visualization software currently takes advantage of the symmetric multiprocessing capability to increase performance. Direct parallel programming of the multiple processors is expected to improve single application performance in the future; for example, partitioning the different cells of the image spreadsheet among the available processors.

Current Low-End and Future High-End Hardware

The state of the art workstations show great performance increases for both the low and high priced workstations. At the low-end a SGI Indigo Elán R4000 RISC "PC", as shown in Fig. 3, is being used to demonstrate the IISS on a less expensive platform.

The Indigo R4000 is currently configured with 144 MB RAM and can be expanded to 384 MB. The backplane data transfer rates on the Indigo are presently four times faster than the more expensive SGI 4D/340 VGX. Image animation rates of 100 frames/s have been demonstrated for 512X512 pixel 8-bit images. The R4000 processor on the Indigo is also about three times faster than the R3000 CPU used in the 4D/340 (85 MIPS and 30 MFlops). However, the Indigo with a 24-bit Elán color graphics board has about half the performance of the high-end VGX graphics board. Currently the local high performance disk storage is only about 2 GB, so most of the image data base must be accessed via Ethernet at about 0.5 MB/s. The Indigo has proven to be an excellent development platform, and has ample performance for modest sized IISS data sets, but cannot be used for some of the high performance IISS demonstrations because of limited local high speed storage facilities.

The high-end workstation graphics can be upgraded to a "Reality Engine" board that has just come on the market. The R4000 multiprocessor CPU and backplane upgrade for the 4D/340 should be available by the time this article is published. The high-end workstation upgrade will have four R4000 processors that benchmark at about 340 MIPS and 130 MFlops. The data paths will be 64 bits wide instead of 32 bits. It is expected that backplane data transfer rates of over 1.0 GB/s will be achieved. The 64 bit architecture will also enable memory configurations of up to 8.0 GB although memory prices are still too high to purchase more than about 1.0 GB. Production of full screen 24-bit animations of 30 seconds and about 5 minutes of broadcast quality digital video will be possible with 8.0 GB of RAM. The Reality Engine supports an increased resolution of 1600 X 1200 and gives a factor of 20 improvement in performance for texture mapped surfaces and complex rendering. Low cost SCSI 2 striped multiple hard disk drives have been benchmarked at the 20 MB/s data transfer rate level.

4. RESULTS

A variety of remote sensing data sets have been used to illustrate some of the IISS features currently available or that are being implemented. The data sets that will be discussed in the following sections include Nimbus-7/TOMS daily and monthly global ozone data for a twelve year period, ERS-1/ATSR data of the Florida peninsula area, and complete Landsat TM scenes of hurricane Andrew.

4.1 Temporal Browse Module Used to Study Polar Ozone Variations Over a 12 Year Period

Figure 4 shows a 3 X 3 cell matrix configuration displaying monthly averages of TOMS ozone data using a colormap to show ranges of Dobson Units (Guimaraes and McPeters, 1991). Eight of the cells contain 12 frames each representing monthly averages for the years 1983 through 1990 while the lower right hand cell has 149 frames of monthly averages for the period spanning Nov. 1978 through Mar. 1991. The IISS allows the independent or synchronized looping and zooming of frames in any combination of cells. In Fig. 4 cells B2 and C3, comparing ozone levels between Oct. 1987 and Oct. 1979 have been grouped together, as indicated by the yellow border, in order to manipulate the two cells in synchrony. The frames in the two cells can be animated, rescaled or enhanced in tandem to examine salient features.

4.2 Thin Cirrus Detection Using Formula Evaluation

Development of a thin cirrus detection algorithm is used to illustrate the IISS formula evaluation capability. Multispectral images from the Along Track Scanning Radiometer (ATSR) on the European Remote-sensing Satellite (ERS-1) for an area off the east coast of South Florida were used to develop and evaluate the algorithm. Difficulties in using a single threshold value for cloud-detection include: (i) high contrast illumination variation across the image, (ii) shadows, (iii) variations in thin cirrus or low convective cloud top brightness temperatures confused with land brightness values; for example, the Florida peninsula area, and (iv) specular reflection areas due to sunglint.

In Figure 5 Cell A1 shows the scene in the 1.6 μm mid-IR region of the spectrum. Cell A2 shows the scene in the 12.0 μm thermal-IR region of the spectrum. A low cloud mask made from the 1.6 μm mid-IR channel is shown in yellow in Cell B1. The ratio of 12.0 μm thermal-IR to 1.6 μm mid-IR was used together with a low cloud mask, as shown in equation (4.1), to give the final result in Cell B2 where the thin cirrus areas are

displayed in red. The expression for detecting thin cirrus clouds is given in (2.3) and shown in the formula panel in Fig. 5 with the result being placed in Cell B2 Frame 1 as specified in the *Frame ID* field. The lower threshold value of 100 for the mask screens out small cumulus clouds especially over land. When the lower threshold value is increased to 125 the edges of small cumulus clouds can also be identified. The same threshold values have been found to be robust for several ATSR images along the same descending orbit over the east coast of the U. S..

$$T_c = \frac{100 \times T_{12\mu\text{m}}}{T_{1.6\mu\text{m}} - \text{Mask}(T_{1.6\mu\text{m}})} \quad (4.1)$$

$$\text{Mask}(T_{1.6\mu\text{m}}) = \begin{cases} 255, & 100 \leq T_{1.6\mu\text{m}}(x,y) \leq 255 \\ 0, & \text{otherwise} \end{cases}$$

$$\text{Thin Cirrus} = \begin{cases} 1, & 75 \leq T_c(x,y) \leq 200 \\ 0, & \text{otherwise} \end{cases}$$

4.3 Hurricane Andrew Landsat TM Image Enhancement Using Khoros Cantata

Landsat 5 TM sensors happened to observe Hurricane Andrew at 10:00 a.m. on August 24, 1992 just a few hours after its devastating passage across the Florida peninsula. For the first time, the fine scale structure of a major hurricane is observed at 30 meter resolution in the 2.2 μm mid-infrared spectral region by Landsat TM channel 5. Since Landsat was designed and calibrated primarily to observe darker land surfaces most of the seven TM channels are saturated by clouds or do not resolve cold cloud temperatures. Even the channel with the best cloud observations required sophisticated image enhancement to bring out cloud details.

Cell A1 of Fig. 6 shows the unenhanced Landsat TM channel 7 image of Hurricane Andrew. The processing steps involved in enhancing the image are shown in Fig. 7 as a Khoros dataflow diagram in the Cantata workspace. Cell A2 of Fig. 6 shows the result of flipping top-to-bottom the image in Cell A1 (*vflip*), then rotating the image by 9.5° counter-clockwise (*vrotate*) so that the sensor scanlines are horizontal, extracting from the resulting image a 980 X 1038 subimage starting at location (143, 84) (*vextract*), and linearly stretching the grayscale range after excluding the upper and lower one percent of outliers (*vhstr*). In Cell B1 the results of a local contrast stretch enhancement using histograms from overlapping 31 X 31 windows (*vhxray*), followed by local averaging using a 3 X 3 window (*vconvolve*) is shown dramatizing the small scale vortex banding structure of the storm. The *vpad* operator is used to add a 15 pixel border to the image to reduce edge effects. The final product is shown in Cell B2 of Fig. 6 where A2 and B1 are blended in an 8 to 2 ratio (*vblend*). The blended image highlights the vortex banding without dominating larger scale features like the eye.

The black-shaded *editimage* glyphs in Fig. 7 correspond to the 2 X 2 image cell matrix of Fig. 6. The *editimage* glyphs in the upper left, upper right, lower left, and lower right represent the unenhanced, linear stretched, local enhanced and final blended images respectively. The *viff2raw* glyphs produce raw image format files of the three output products.

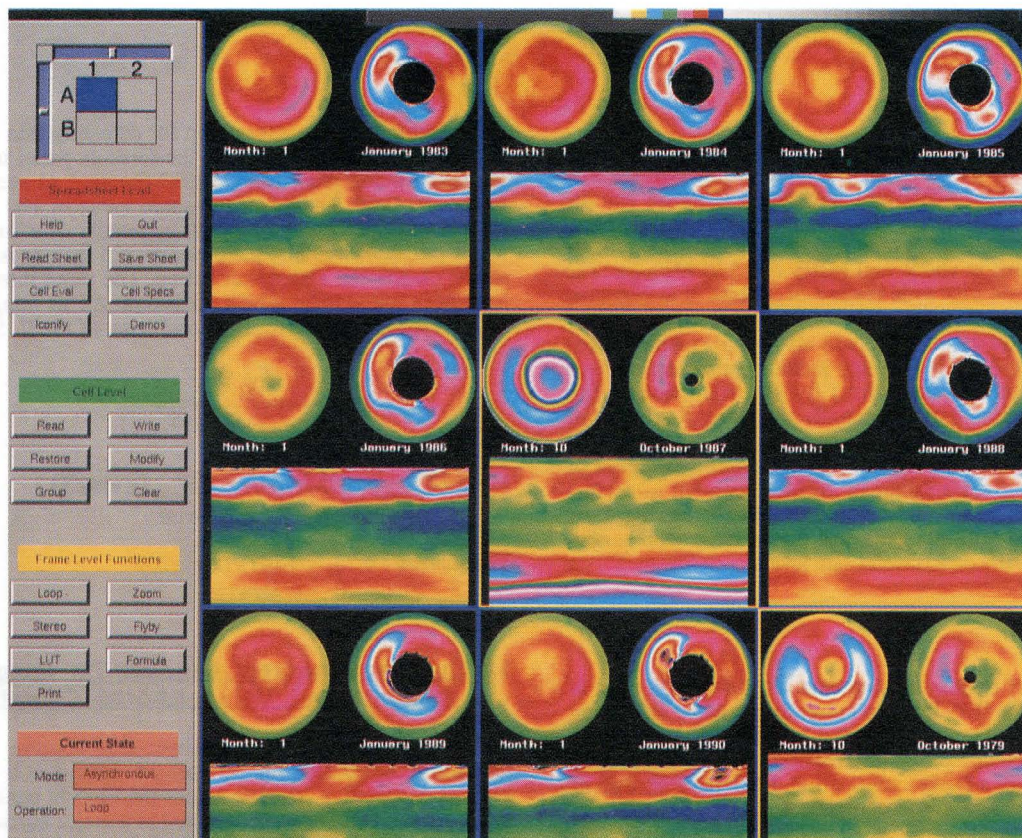


Figure 4. A 3 X 3 cell matrix, with the IISS control panel on the left, showing monthly averages of TOMS ozone data.

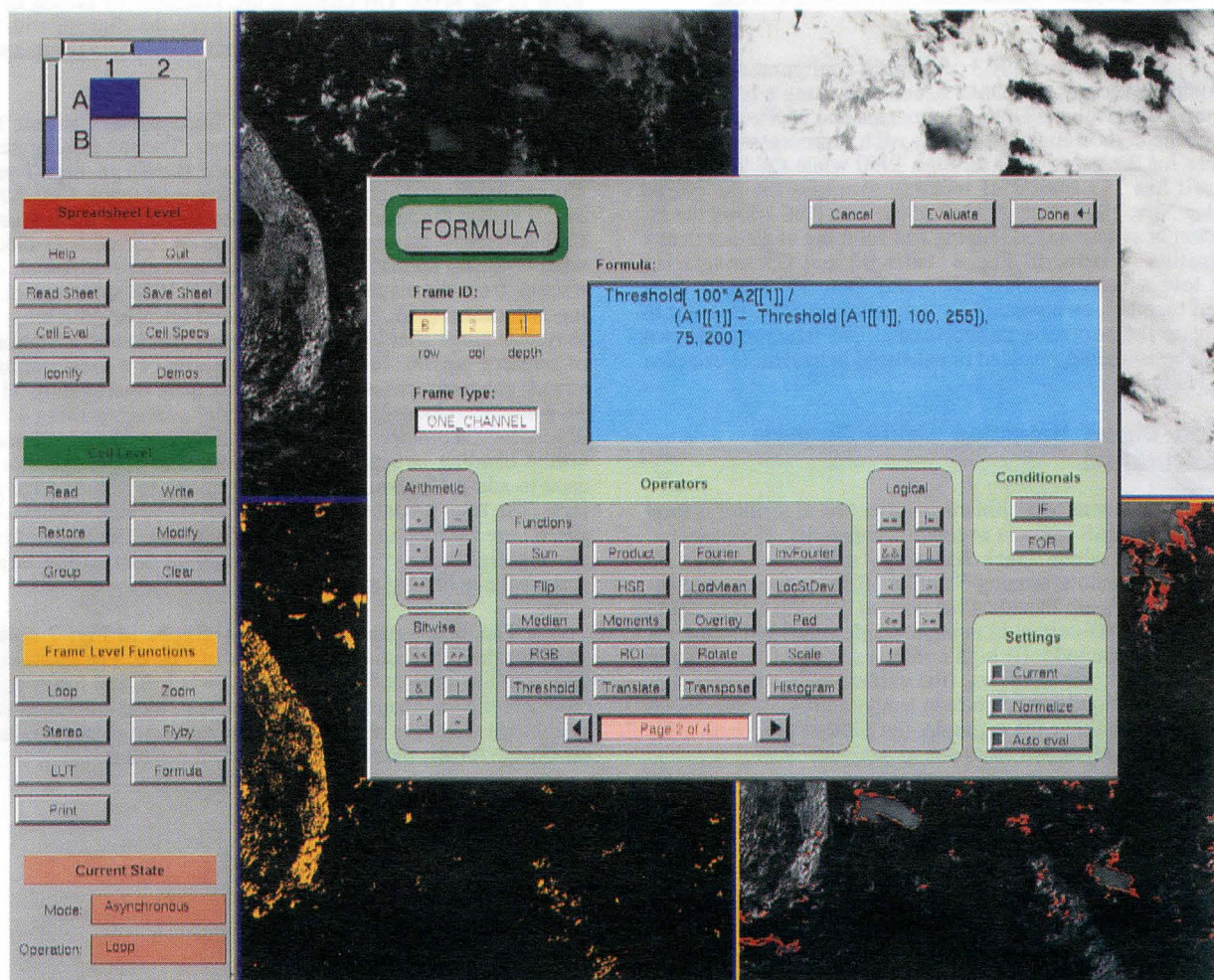


Figure 5. ERS-1/ATSR images and results of thin cirrus detection algorithm showing formula evaluation panel.

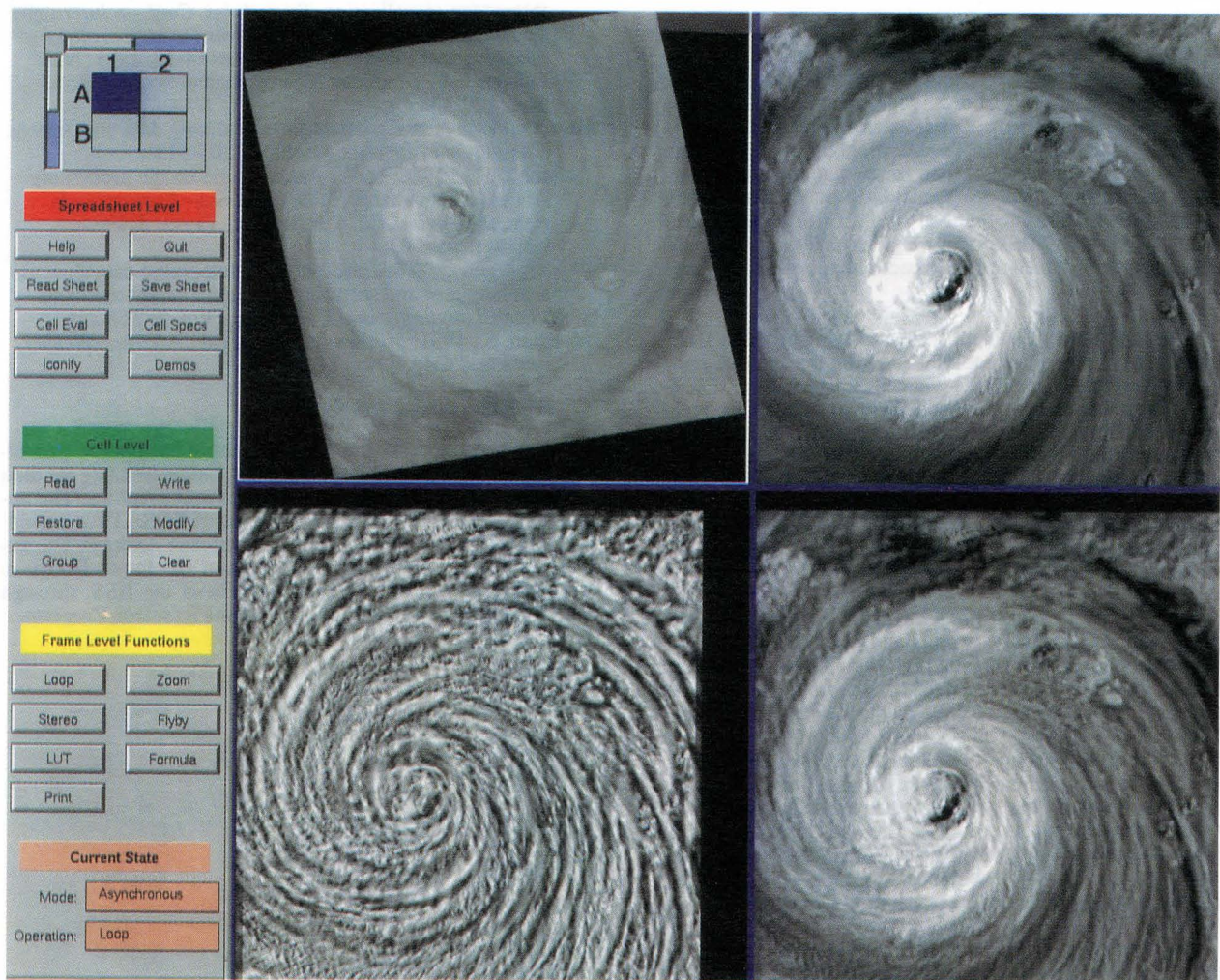


Figure 6. Multistage enhancement of an image of Hurricane Andrew as observed by Landsat 5 TM channel 5. See text for details.

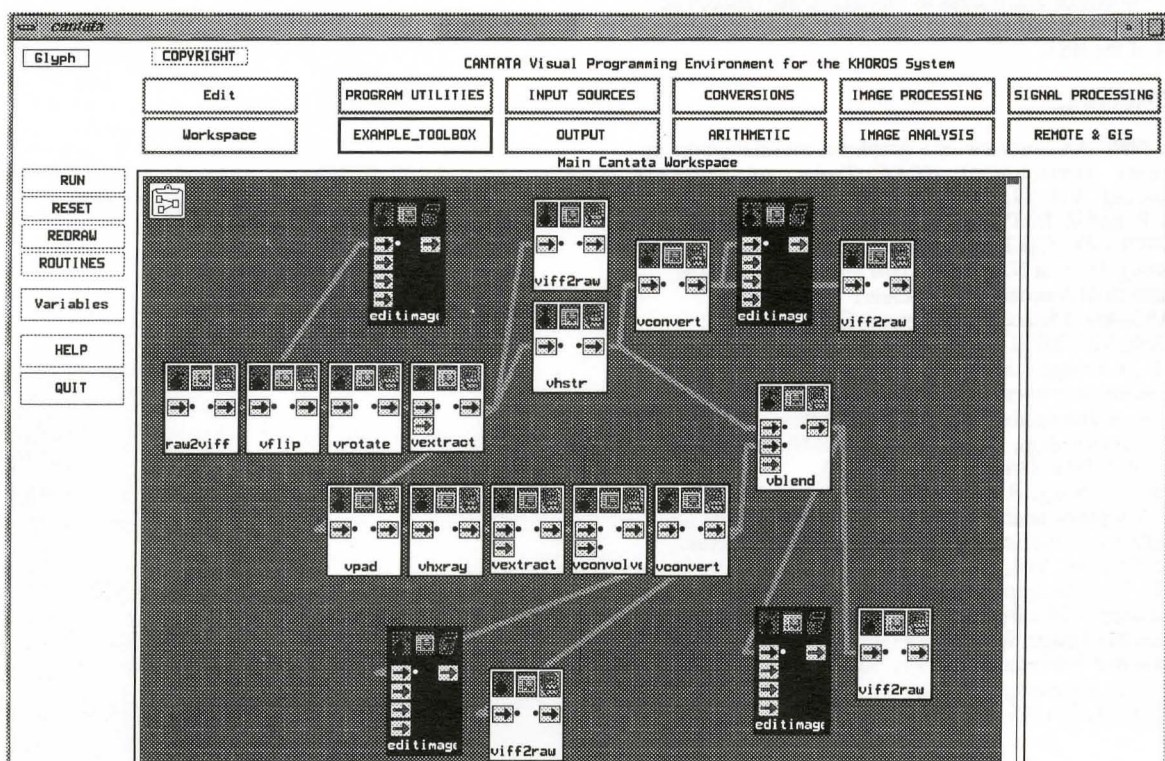


Figure 7. The algorithm used to process Cell A1 in Fig. 6 is shown using a dataflow diagram in the Cantata workspace of Khoros.

5. CONCLUSIONS AND FUTURE WORK

The Interactive Image Spreadsheet (IISS) environment has been shown to be a useful tool for developing quick exploratory image analysis algorithms and for browsing through large databases. The temporal browse feature for rapidly examining large volumes of data was illustrated using the twelve year TOMS ozone data set. Although monthly averages were shown, the daily ozone data set can be used and moving averages over different time windows calculated interactively using the formula evaluation mechanism in the IISS. Other useful tools provided by the IISS for "mining" large image databases include high performance spatial, spectral and volume browsing.

The exploratory development of deriving products from satellite imagery was illustrated by showing the relative ease with which a thin cirrus detection algorithm using ERS-1/ATSR data could be constructed. The powerful formula evaluation tool enables the user to construct algorithms and evaluate performance without having to worry about the details of input and output operations or programming in a low level language. The necessity to generate a large number of intermediate image products in order to evaluate and compare the strengths of different algorithms is facilitated by the 2-D cell and 3-D frame-stack organization of the IISS.

The ability to visualize algorithms that may involve complex relationships between cells and frames in the image spreadsheet was illustrated using the Khoros Cantata workspace. Formulas constructed using the spreadsheet syntax are parsed into a sequence of calls to the Khoros image and signal processing library. The sequence of Khoros calls map directly into a dataflow diagram with glyphs (nodes) representing image processing modules and arcs connecting the glyphs specifying the flow of data as well as the parallel nature of the algorithm. The enhancement of a Landsat TM image of Hurricane Andrew to bring out small scale cloud structure was used to illustrate the relationship between the IISS formula evaluation capability and the Khoros dataflow paradigm.

The current high performance hardware environment has made possible the effective manipulation and analysis of moderately large satellite data sets. It is anticipated that future improvements in CPU performance, disk i/o performance, network performance and access to multigigabyte random access memory will be necessary to fully take advantage of the capabilities of the IISS.

6. REFERENCES

- Fisher, G., 1988: An overview of a graphical multilanguage applications environment. *IEEE Trans. on Software Engineering*, Vol. 14, No. 6, 774-786.
- Guimaraes, P. and R. McPeters, 1991: *TOMS Ozone Image Data 1978-1991*, CD-ROM prepared by the Ozone Processing Team at NASA/Goddard Space Flight Center. Available from National Space Science Data Center, NASA/Goddard Space Flight Center, Code 633, Greenbelt, MD 20771.
- Hasler, A. F., J. Dodge, and R. H. Woodward, 1991a: A high performance interactive image spreadsheet. *Seventh Int. Conf. on Interactive Information and Processing systems for Meteorology, Oceanography and Hydrology*, Jan 14-18, 1991, New Orleans, LA, 187-194.
- Hasler, A. F., J. Strong, R. H. Woodward, and H. Pierce, 1991b: Automatic analysis of stereoscopic image pairs for determination of hurricane cloud top height and structure. *J. Appl. Meteor.*, Vol. 30, 257-281.
- Hasler, A. F., K. Palaniappan and D. Chesters, 1992: Visualization of Multispectral and Multisource Data Using an Interactive Image Spreadsheet (IISS). *Eighth Int. Conf. on Interactive Information and Processing Systems for Meteorology, Oceanography and Hydrology*, Jan. 5-10, 1992, Atlanta, GA, 85-92.
- Hibbard, B. and D. Santek, 1991: The VIS-5D system for easy interactive visualization. *Seventh Int. Conf. on Interactive Information and Processing systems for Meteorology, Oceanography and Hydrology*, Jan 14-18, 1991, New Orleans LA., American Meteorological Society, Boston, MA, 129-134.
- Hon, D., 1991: A new method for visualizing data on a sphere. *Computers in Physics*, Sept/Oct, 505-513.
- Jackson, P. H., 1988: IAX - An algebraic image processing language for research. In *Digital Image Processing In Remote Sensing*, Ed. J. P. Muller, Taylor & Francis, London, 79-103.
- Rasure, J. and C. Williams, 1991: An integrated data flow visual language and software development environment. *Visual Languages and Computing*, Vol. 2, 217-246.
- Wolfram, S., 1991: *Mathematica*, 2nd Ed., Addison-Wesley, Redwood City, CA.

7. ACKNOWLEDGMENTS

The authors wish to thank Dennis Chesters for providing the TOMS data, and Hal Pierce and Lafayette Long for their assistance in the preparation of images and graphics. The authors also wish to thank Mark Emmons of EOSAT for providing timely Landsat TM imagery of Hurricane Andrew. The authors appreciate the continued support of Dr. James Dodge in the conceptualization and development of the IISS, through NASA's Energy and Hydrologic Systems Research Program.