Reprinted from

Tenth International Symposium

Machine Processing of

Remotely Sensed Data

with special emphasis on

Thematic Mapper Data and

Geographic Information Systems

June 12 - 14, 1984

Proceedings

Purdue University
The Laboratory for Applications of Remote Sensing
West Lafayette, Indiana 47907 USA

A DEVICE - INDEPENDENT INTERFACE FOR IMAGE DISPLAY SOFTWARE

D.C. PERKINS, M.R. SZCZUR,
J. OWINGS, R.K. JAMROS

National Aeronautics and Space
Administration/Goddard Space
Flight Center
Greenbelt, Maryland

NASA's Goddard Space Flight Center (GSFC) has developed a Transportable Applications Executive (TAE) for use in implementing portable applications software that can be shared by different research projects. Since many of the supported disciplines require the interactive display and manipulation of remotely sensed images, a device independent Display Management Subsystem (DMS) is being written as a TAE extension. The DMS attempts to abstract and standardize the device dependent functions that are used in the display and manipulation of image data on image analysis terminals. This paper explores the structure of DMS and the interface routines that are available to the applications programmer for use in developing a set of portable image display utility programs.

I. INTRODUCTION

Much of the research carried out at NASA's Goddard Space Flight Center (GSFC) involves computer-aided analysis of remotely sensed image data obtained by meteorological, oceanographic, or earth resources satellite missions. The scientific analysis and interpretation is aided by the ability to display and manipulate the data quickly and easily using raster devices, referred to here as image analysis terminals (IAT).

The rapidly changing market in IATs over the last decade, combined with disparate system development cycles and goals, has led to a proliferation of analysis systems configured around different IATs. The result of such a history of development is a collection of programs, tailored to specific IATs, which cannot be used on different IATs without spending considerable time and effort on conversion. Yet, many of the programs could otherwise be usefully shared among the different systems.

The Display Management Subsystem (DMS) of the Transportable Applications Executive (1) was conceived as a means of standardizing the interface to a variety of IATs in support of new projects. It offers device independence so that programs can be made portable among a variety of IATs. While it is a subsystem of TAE, the concepts and techniques are general and could be readily implemented outside of TAE.

II. DMS CONCEPTS

DMS has two primary objectives: to establish a software environment for an interactive user that allows that user to control, manipulate and do analysis using an IAT without having to understand specific characteristics of the hardware or supporting software; and to allow programs which access IATs to be written independent of any specific IAT type.

To meet these objectives, the DMS designers established three major requirements:

1. users of DMS must have services, similar to operating system file management services, for managing data displayed on an IAT;
2. programs must be able to perform actions on IATs without addressing a specific vendor's hardware characteristics;
3. an interactive user must be able to exclusively control or selectively (and deliberately) share a particular IAT, and have that device used automatically by programs he/she runs.

In support of these requirements, the DMS designers formalized definitions of "images" and IAT categories for DMS users.

A. IMAGES

In an interactive image display and processing system, the data base that is operated on by the software consists of digital image files and related ancillary information. These files are transferred to display devices for viewing.

In a departure from earlier systems (e.g., references 1, 2), which required that a user first place data into refresh memories, then independently apply intensity transformation (lookup) tables, configure the memories for viewing, and finally, remember the details of this configuration for later viewing, DMS combines these types of sequences into single operations. The result of an operation is named, and the entire configuration can be later recalled by use of that name. So, for example, in one operation a user can load three bands of data, declare them to be a false color image, and name that image WASHINGTON. Later, after displaying other data sets or results, the user can type

VIEW WASHINGTON

to return to viewing the original false color image.

B. IAT CATEGORIES

DMS operations are based on the concept of generic devices. To DMS, any one IAT is identified only as a collection of capabilities for data storage and control. For example, a simple device may be identified as having three 512 X 512 refresh memories, three lookup tables, one monitor, one cursor, and one graphics overlay plane.

Each of these collections of characteristics may be given one or more names. The purpose of the naming is twofold. First, IATs can be given meaningful names, which will allow users to allocate them simply. Second, programs may also request IATs by name. DMS requires that a program which accesses an IAT declare what capabilities it expects to use. This powerful feature allows programs to be concerned only with what they intend to accomplish, not with specifics of a device. DMS verifies this request against the capabilities of the IAT owned by the user of the program.

III. DMS STRUCTURE

DMS is founded upon a set of data structures which capture and identify the main DMS concepts. Several software components have been developed to manipulate the data structures and to allow programmer access to the IATs. These include:

- o Data structure management
- o Generic image manipulation services
- o Device dependent services
- o Image input/output support
- o Image display utility programs

The following sections describe the characteristics and usage of the various components, followed by specific software implementation details.

A. DMS DATA STRUCTURE MANAGEMENT

Four data structures support all environmental and control information maintained by DMS:

Device ID Block. This block identifies the IAT currently owned by a user.

Display Memory Table. The DMT associates images defined by a user with a device's memories and maintains characteristics of all images, e.g., image type, name, physical location(s), age, lock switch, etc.

Display Device Table. The DDT contains a physical description of IAT characteristics (e.g., number of refresh memories, number of cursors, number of buttons, other hardware features) plus user information (e.g., name of IAT, current owner).

Device Name Table. The DNT contains all names by which any IAT on a given system can be known. IATs may have more than one name, and names may be generic (e.g., monocolor device), specific (e.g., XYZ 60) or facility-inspired (e.g., TIGER).

B. GENERIC IMAGE MANIPULATION SERVICES

Application programs interact with IATs through a set of generic DMS image manipulation functions, collectively known as the XD package. Use of this package for all device interaction makes the program usable on any IAT within a system.

The XD package serves the purpose of hiding all DMS table manipulation and physical device access from the application program.

DMS also handles the windowing of IAT-resident images at the XD level. Incoming data may be written into the refresh memory at an offset from the base coordinates of that memory. Any subsequent references to points in the resulting image are made relative to the beginning of the image. That is, a user works in the coordinates of the image; DMS translates between that system and refresh memory coordinates.

C. DEVICE DEPENDENT SERVICES

The DMS software layer which directly addresses an IAT is known as the DD package. Where the XD package is portable across IATs, the DD package must be reimplemented for each different IAT architecture, and thus serves as a "device driver". Unlike the XD routines, which deal in "images" and windows, DD routines work with refresh memories and other hardware elements and access an IAT directly through vendor supplied or other device specific software.

D. IMAGE I/O SUPPORT

DMS was targeted for use on several systems, each of which had its own disk-based data structures for images. Rather than try to accommodate system-specific formats, the DMS team developed a set of protocols for accessing image data which could be layered over locally used I/O, and which were independent of any physical data structure. (2)

This package allows an application program to read, write and update image files and their labels. In its initial implementation, it requires that an image file be a single band image stored as a disk file. Subsetting of a disk-based image for input is supported. Single refresh memories can be used as an input or output image.

The I/O routines maintain their independence of data structure by using keyword parameters to describe the data. Thus, when a program needs a particular kind of information, such as pixel size, it requests it by keyword. The implementation of this package and its tables are system dependent. The calls are generic. Use of these routines will facilitate the porting of programs to other systems which also have an implementation of this package.

E. UTILITY PROGRAMS

A set of image display utility programs are provided along with the DMS. These programs serve two purposes. They do many simple image display operations, making it unnecessary for each installation to code them. They also serve as a model of the application interface to DMS for the discipline specific programs that are developed by each site. The following lists many of the DMS-supplied programs.

- ALLOC - allocate an IAT to a user session
- DEALLOC - free a previously allocated IAT
- IATINIT - set the IAT and DMS tables to a known initial state
- IATSTAT - list capabilities, status of system IATs
- TOTV - create an image on an IAT from disk image file(s)
- FROMTV - create disk image file(s) from an IAT image
- IMGLST - list current images with their attributes
- IMGUTIL - update image list
- VIEW - display an image on a monitor screen
- LOOP - show a sequence of images
- ALIGN - interactively register images by shifting
- FADE - interactively fade between two images
- SPLIT - display portions of images simultaneously
- ZOOM - enlarge an image area by pixel replication
- PAINT - dynamically make assignments to grey levels
- STRETCH - make LUT assignments for contrast alteration
- HIST - compute the intensity histogram of an image
- PROFILE - list the intensity values along a line
- LOADLUT - load lookup tables to the IAT
- SAVELUT - save lookup tables to disk

F. IMPLEMENTATION

A prototype DMS has been implemented on a VAX 11/780 under VMS. The primary implementation language for table manipulation and image I/O is C, while FORTRAN 77 is used for other subroutine packages. All routines for application programmers are FORTRAN-callable.

Device independence is attained through layering the software into link and run time libraries. Application programs link to the generic package. The selection of the device-dependent package to be used (i.e., the mapping to a particular IAT) is done at run time, based on a user's allocated IAT. Under VMS, DMS uses sharable libraries to provide the run time library linking. The DMS control tables are stored in a global section.

Detailed DMS documentation (Functional Specifications, Applications Programmer's Guide, System Programmer's Guide) is available from the authors.

IV. SUMMARY

Advancing technology and increasing maintenance costs on older equipment will continue to make system upgrades necessary. In particular, more powerful and less expensive image analysis terminals are being marketed by a variety of vendors. At the same time research laboratories like GSFC have invested tens of man years in program development for applications that are not commercially available. A great deal of effort has been expended in making the user interface to these programs a "tool that fits the hand" of the scientist. To preserve the unique algorithms and familiar user interfaces, difficult and expensive (and boring) conversions are required to add new IATs to an existing facility.

The DMS is an attempt at making these changes easier than in the past. The requirement to make the software more portable among disparate image devices is basic to the design of the DMS. In particular, the DMS distances both the end user and the applications programmer from the IAT hardware. We are persuaded that the systems we build based on the DMS will be more portable, easily maintained, and usable than without it.

The DMS prototype was completed in the Spring of 1984. Its functional capabilities are those discussed in this paper. Some areas of refinement for development of a mature DMS are being explored. To date, these include expanding control for the display of data (e.g., managing viewports); generalizing the concept of image type, for example, to support n-band images associated with arbitrary transformation tables; supporting the naming of groups of images, for example, for loop sequences, mosaics, or images combined through boolean operations; supporting new functions such as creation of perspective images, image rotation, and Fourier transforms; and formalizing the relationship between DMS and graphics packages.

V. ACKNOWLEDGEMENTS

DMS is being developed by the Advanced Systems Development Branch of the NASA/Goddard Space Flight Center. The work is sponsored by the NASA Information Systems Office, which is part of the Office of Space Science and Applications.

The first use of DMS is in the upgrade to the Atmospheric and Oceanographic Information Processing System being developed for the Goddard Laboratory for Atmospheric Sciences. The EROS Data Center of the U.S. Geological Survey and the upgraded GSFC Land Analysis System (LAS) will also be DMS users.

VI. FOOTNOTES

1. The transportable Applications Executive was developed by the NASA/Goddard Space Flight Center's Advanced Systems Development Branch to provide a standard and portable interface for users of scientific research and analysis systems (reference 3).

2. Early design work on the image I/O package was done with the Jet Propulsion Laboratory under a collaborative software development agreement between the JPL Multimission Image Processing Laboratory and the TAE project.

VII. REFERENCES

1. Bracken, P.A., J.T. Dalton, J.B. Billingsley, J.J. Quann, "Atmospheric and Oceanographic Information Processing System (AOIPS) System Description", NASA X-933-77-148, March, 1977

2. Dalton, J.T., R.K. Jamros, D.P. Helfer, D.R. Howell, "The Visible and Infrared Spin Scanning Radiometer (VISSR) Atmospheric Sounder (VAS) Ground Data Systems", Society of Photo-Optical Instrumentation

Engineers Technical Symposium, April, 1981

3. Helfer, Dorothy P., David R. Howell, Jan Owings, Martha R. Szczur and John T. Daltón, "A Transportable Executive for Interactive Applications", Proceedings of the Harvard Computer Graphics Conference, July, 1981

VIII. AUTHOR BIOGRAPHICAL DATA

Dorothy Perkins is a software project leader in the Advanced Systems Development Branch, Information Extraction Division, NASA/GSFC. She was a member of the TAE design team and was the project manager for the TAE implementation. Currently, she is working on the upgrade of the AOIPS system and has been a key person in the definition of the requirements of the DMS. She received a B.A. in Mathematics from Wellesley College and is a member of the ACM and the IEEE.

Martha Szczur is a computer scientist in the Advanced Systems Development Branch. She received a B.A. in Mathematics from Converse College and is a member of NCGA and ACM. Her experience includes work on the design of the TAE and image display support packages for the LAS and DIDS (Decision Information Display System). At present she is the DMS project manager.

Jan Owings is a systems analyst in the Advanced Systems Development Branch. She received a B.A. in Mathematics from Winthrop College. A member of the TAE design team, she has worked on astronomical and earth remote sensing image processing and display systems. She is now a member of the DMS design team.

Rita Jamros is a systems analyst in the Advanced Systems Development Branch. She received a B.A. in Mathematics from St. Francis College. She currently is a member of the DMS design team. Her previous experience includes work on the design of the TAE and on the VAS ground data system.