A Methodology for Evaluation of an Interactive Multispectral Image Processing System

William M. Kovalick and *Jeffrey A. Newcomer*
Science Applications Research (SAR), 4400 Forbes Blvd., Lanham, MD 20706
Stephen W. Wharton
NASA Goddard Space Flight Center (GSFC), Greenbelt, MD 20771

ABSTRACT: Because of the considerable cost of an interactive multispectral image processing system, an evaluation of a prospective system should be performed to ascertain if it will be acceptable to the anticipated users. Evaluation of a developmental system indicated that the important system elements include documentation, user friendliness, image processing capabilities, and system services. The criteria and evaluation procedures for these elements are described herein. The following factors contributed to the success of the evaluation of the developmental system: (1) careful review of documentation prior to program development, (2) construction and testing of macromodules representing typical processing scenarios, (3) availability of other image processing systems for referral and verification, and (4) use of testing personnel with an applications perspective and experience with other systems. This evaluation was done in addition to and independently of program testing by the software developers of the system.

INTRODUCTION

THE SCARCITY of literature on general capabilities of multispectral image processing systems (Bracken, 1983) requires that potential users of such systems must develop *ad hoc* criteria for evaluation and verification. The problem with this approach is that users with limited experience may be unaware of the need to consider specific system elements in addition to the image processing functions. These elements include documentation, system services, and user friendliness. The implication is that, without such information, a system with inherent limitations may inadvertently be accepted.

In the context of this paper, a multispectral image processing system refers to a software based system used for processing of digital multispectral data acquired from satellite and aircraft platforms. The purpose of this paper is to provide a methodology for evaluation of such a system to help ensure that the system contains the desired user features. The system elements believed to be critically important are noted and specific criteria for their evaluation are given. Procedures for testing these critical system elements are also given.

Image processing systems typically consist of the actual image processing software integrated with a number of ancillary subsystems. The authors acknowledge the importance of ancillary subsystems to perform statistical analysis, geographic data entry and analysis, storage and retrieval of tabular data by means of data base software, and possibly for other purposes. Due to the diversity of capabilities contained in ancillary subsystems, discussion of their evaluation is properly the subject of separate publications. We also believe that the main priority is to ensure that the main system is adequate before considering any subsystems. For these reasons, evaluation of ancillary subsystems is not addressed in this paper.

The need for a systematic approach to evaluate an image processing system became apparent during the development of the Land Analysis System (LAS) at the National Aeronautics and Space Administration (NASA) Goddard Space Flight Center (GSFC). The LAS was developed to satisfy the image processing needs of scientists conducting research in forestry, agriculture, geology, and hydrology within the Laboratory for Terrestrial Physics (LTP) at GSFC. The overall systems and functional requirements for the LAS were derived on the basis of prior experience with existing image processing systems. Criteria and testing procedures were developed to ensure that the LAS satisfied the LTP requirements. This paper is based on the experience of the authors in testing the LAS.

EVALUATION CRITERIA

A review of image processing requirements indicated that separate evaluation criteria were required for each of four distinct system elements. These elements, which were considered to be of critical importance for any image processing system, are (1) documentation, (2) user friendliness, (3) applications functions, and (4) general system services. The evaluation criteria for these elements are described in detail.

DOCUMENTATION

Although users may have a tendency to initially ignore documentation, adequate documentation can serve to minimize user errors caused by a lack of information and to minimize the amount of training needed for new users. The documentation may be in hardcopy form in manuals and/or online (on the computer). If both forms exist, the two forms should be consistent to alleviate user confusion. Experience has shown that the availability of both forms is desirable. A favorable aspect of hardcopy documentation is that it does not require computer system resources for review. A benefit of online documentation is that it provides the user with a quick reference without interrupting a processing session by having to search for hardcopy documentation.

The documentation should contain an overview of the system and detailed descriptions of the individual applications functions. Features that should be included in the system overview and individual function documentation are given in Tables 1 and 2. User-modifiable parameters (Table 1) refer to parameters such as those that affect (1) the enabling/disabling of specific hardware devices (e.g., array processor), (2) the enabling/disabling of session logging, (3) the specification of a standard output device,

TABLE 1. COMPONENTS OF SYSTEM OVERVIEW DOCUMENTATION

- A synopsis of the required system hardware
- An overview of the software packages, capabilities, and interfaces
- How to logon and start to use the system
- Use of user-modifiable parameters affecting overall processing
- Command syntax
- Modes of system operation (interactive, batch, asynchronous)
- System processing limits (e.g., maximum number of bands)
- File types, characteristics, and naming conventions
- Applications programmer's reference guide

0099-1112/87/5308–1087$02.25/0

TABLE 2. COMPONENTS OF APPLICATIONS FUNCTION DOCUMENTATION

- Function name and a brief description of the processing performed
- Processing limitations (e.g., number of bands, number of classes)
- Parameter names, descriptions, default and valid values
- Input and output image data types
- Examples of running the function
- Processing times for typical execution of the function
- Detailed description of the algorithm being used
- Error messages and appropriate user responses
- Information providing insight into using the function
- Listing of hardware used
- Reference material
- Date the function was released

TABLE 3. USER FRIENDLINESS CRITERIA

I. System Level Criteria
- Recall of previous commands and parameter values
- Compatibility of inputs and outputs between functions
- Capability to halt a currently executing function without causing damaging side effects

II. Applications Function Criteria
- Processing status messages for individual functions
- Identical names for the same parameter in each function
- Meaningful parameter names or mnemonics
- Consistent user responses (e.g., affirmative responses should always be YES or Y)
- Specifiable spatial and spectral subsets
- Proper file handling upon abnormal function termination (e.g., deletion of intermediate or incomplete files)
- Software verification of sufficient system resources (e.g. disk space) before attempting processing
- Validation of parameter values upon entry
- Reprompting for erroneous parameter values
- Minimization of parameter to parameter dependencies (e.g., value of one parameter should not affect the meaning of another parameter)
- Option of sending results to terminal, disk file, line printer
- Minimization of user steps for tape operations

and (4) the specification of software libraries to search for applications functions. The effect of user-specifiable system parameters on individual applications functions needs to be included in the individual applications function documentation. For example, a function may have both an array processor and a host computer version, but the array processor version does not have the flexibility to process images of a given size. Omittance of such seemingly insignificant information can cause a great deal of user frustration.

USER FRIENDLINESS

If a sufficient degree of user friendliness is present, the first-time user as well as the experienced user should be able to use the system effectively. For example, the availability of a menu mode for entering parameters and viewing their descriptions is particularly beneficial to a novice user. An experienced user, however, may prefer to operate in a more efficient command mode in which functional names and parameter values are entered on a single command line. The availability of both modes of operation is, therefore, desirable. Experience has shown that it is better to have more user friendliness than is thought to be needed because it seems that the software is never quite as friendly as it should be. Overall, user friendliness encompasses items such as (1) proper error handling by software, (2) consistent interfacing of the programs with the user, and (3) smooth interfacing of the programs themselves. Specific facets of user friendliness are presented in Table 3.

Particular attention should be given to evaluation of parameter input because this is the primary manner in which the user interacts with the applications functions. If a given function is run repeatedly or if it has a large number of parameters and it is run more than once, it is very convenient to be able to recall previously entered parameter values. If batch processing is available, the means for entering parameter values is also important. The preferable way to accomplish this is by specifying a parameter name and associated value rather than sequential interactive prompting because the interactive prompting cannot be done in batch processing.

IMAGE PROCESSING CAPABILITIES

A review of existing image processing systems reveals that certain basic processing capabilities are contained in them (Spencer, 1974; Haralick *et al.*, 1977; McMurtry *et al.*, 1977; ESL, 1983; Greenlee and Wagner, 1982). These capabilities include general data manipulation, data transformation, image enhancement, geometric transformation, convolution, radiometric correction, and pattern recognition. Aside from insuring that the desired functionality is present in a system, a thorough evaluation should also examine the way in which the functionality is provided. For example, a system may provide the needed pattern recognition algorithms but the algorithms may not (1) handle a sufficiently large number of classes, (2) provide acceptable processing speed, or (3) provide sufficient precision for output of statistical information. Therefore, criteria should be constructed to evaluate processing limitations, processing efficiency, and output results.

Typical processing limitations include the data types (byte, integer*2, etc.), image sizes (number of pixels, number of lines), number of bands, and number of categories or greylevels handled by the software. Functions that manipulate discrete (integral) data should allow processing of signed integer as well as byte data (256 values), where possible, to allow increased flexibility. Also, algorithms that perform image classification should be able to process floating point data so that variables such as texture features and geophysical data can be used directly. At a minimum, the software should be able to process images that are 512 pixels by 512 lines because most displays can handle this size. The number of image bands that the system should be capable of handling is dependent on the types of imagery planned for use. With 128-band instruments currently in use (Goetz *et al.*, 1985) and instruments with more bands anticipated (Vane *et al.*, 1984), due consideration needs to be given to the number of bands in order that anticipated sensor data are not precluded from use on the system.

The computational processing speed of the applications functions must be tested, though it may be difficult to establish acceptance criteria. If a second image processing system is not available as a reference, one must rely on intuition and user feedback. If a given function is too slow, a determination should be made as to how frequently the function will be used. If used infrequently, the slow speed may be tolerable. If used frequently, then the other aspects of the system as a whole need to be considered and weighed against other possible candidate image processing systems. If the system is a developmental one, review of the actual software code may be done to identify and, one hopes, correct any obvious inefficiencies.

In evaluation of output results, the primary concern is that the results are correct. Additionally, output results should be properly formatted based on the output device and the data type of the values. Because the space on a terminal screen is limited, the software should provide an expanded format for line printer output. Rather than using one format for all output results, the data type and expected range of the values should determine the format used. For example, an integer format should be used for integer values, instead of a floating point format which may truncate important digits.

SYSTEM SERVICES

System services as defined here include handling of data on tape, display of image data to a video display, data set cataloguing, a means of documenting the processing steps used, and programming environment. These services are essential to efficient image processing and, therefore, warrant thorough evaluation.

Tape Input/Output Utilities. The first step in image processing is to enter the data into the system and the last step is often the removal of the processing results from the system. This typically necessitates the input and output of data from and to magnetic tape. The capabilities required to handle tapes include (1) ingestion of various tape formats including image and support data sets, (2) output of data in various formats, and (3) tape analysis utilities. Each of the numerous tape formats, e.g., Universal, EDIPS (EROS Digital Image Processing System), TIPS (Thematic Mapper Image Processing System), Telespazio, ASCII, and EBCDIC encoded support data, has a unique logical record, physical record, and label format. Hence, it should be known in advance what forms will generally be processed. When dealing with large data sets such as those from the Landsat Thematic Mapper, some image processing systems do not have the capability of storing the entire data set on disk. Therefore, it is important that the tape input/output software be very flexible and allow handling of subsets of full data sets.

The subsetting capability should allow spatial subsetting, band selection, and the use of pixel and line increments to subsample the data. Other features that are highly desirable for tape input and output are (1) availability of several tape storage densities (primarily a hardware issue), (2) the ability to skip label records and files, (3) handling of different data types and machine formats, (4) the capability of handling variable record blocking factors, and (5) handling of band-by-band and line-by-line band organizations.

The tape handling services of an image processing system are strengthened by the existence of tape utility functions. These utilities should allow (1) determination of physical record and file sizes on tape; (2) dumping contents of tape records and files in various formats (hexadecimal, octal, ASCII, EBCDIC); and (3) copying from tape to tape, tape to disk, and disk to tape.

Display. Items to be considered in evaluating a display system include (1) the amount of data the display can handle, (2) the data types that can be displayed, and (3) the actual manipulation capabilities of the display.

Currently, the majority of image displays are capable of displaying three bands of image data with up to 512 pixels by 512 lines per band (Bracken, 1983). Some displays are available that can accommodate three image bands with 1024 pixels by 1024 lines per band. The minimum number of memory planes a display system should have is three. However, six planes can make a display system much more efficient to use. The availability of three planes for storing a color overview of the area of interest and three planes for use in detailed viewing, roaming, and graphics overlaying greatly facilitates image analysis tasks.

Most interactive video display systems handle byte data only. Display of other data types requires some type of scaling operation. Preferably, the display software should allow the user the option of determining if and how the scaling is done. In specifying data to be displayed, it is important that the capability exists to display selected image bands and areas. If the area is too large for the display, the display software should allow pixel and line increments to be applied to get an overview of the area.

The fundamental image manipulation capabilities that a display should provide include intensity mapping, graphics overlaying, roaming at various scales, and coordinate and pixel value determination. As important as this functionality is, it is equally important that the interfacing between display functions be smooth. The need for this is exemplified when an attempt is made to display an image area, zoom a smaller area, modify the intensity mapping, save the intensity mapping for later use, outline a polygon or mark a point on the zoomed area, and then view the results on the original image. Experience has shown that each of these operations should exist as a separate function that can be accessed at any point during the display manipulations. The flexibility this provides is preferable to combining many capabilities into one large program. To have each of the basic capabilities as individual functions does, however, require that a record of the current display status be kept and be properly updated. Once the display operations have been completed, the capability should exist to save image display and graphic results in image format on disk for later use.

Data Set Cataloguing. An efficient means of managing data files without requiring the user to possess detailed knowledge of the host file management system is the basic purpose of catalog management software, herein indicated as a "catalog manager." A catalog manager should provide basic utilities to (1) list files by name and/or attribute, (2) delete files, and (3) rename files. Additional capabilities include (4) the creation of associations between files, (5) file referencing using a range of attribute values, (6) assigning abbreviated auxiliary names to files, (7) restoration of mistakenly deleted tape files, (8) condensing of active files in the tape library, and (9) storage of ancillary information such as file type, image size, datatype, number of bands, etc.

A well designed catalog manager allows access to files by file names and/or file attribute specifications. For example, it is convenient for all files created for a particular project to be accessible by providing the project name rather than specifying all the individual file names. At a minimum, the user should be able to access a group of files that contain a common sequence of characters in their names. Regarding file names, a catalog manager should allow specification of longer, more descriptive file names than does the host operating system.

When properly integrated with a catalog manager, a tape library allows easy access to offline as well as online files. Rather than having to specify all the characteristics of a file on tape (file number, file size, etc.), the user simply needs to specify the file name. Furthermore, a tape library enables convenient storage of large amounts of data that would otherwise restrict available disk space while still keeping data sets easily accessible.

Processing History. Experience has shown a processing history to be invaluable in retracing the steps performed to arrive at a given set of results. Two separate processing histories should be maintained: a session history and an image history. Desirable characteristics of both types are discussed below. As a reference, the current Interactive Digital Image Manipulation System provides the bulk of the session and image history capabilities that follow (ESL, 1983).

A session history contains a record of processing performed and results obtained during a given processing session. Specific information that should be included in a session history are (1) the date, (2) function names, (3) parameter names and values, (4) error and informational messages, (5) processing results that are normally displayed at the user's terminal, (6) completion messages, and (7) a time stamp or a number for each entry. Optimally, the generation and subsequent listing of the session history are under the user's control. This encompasses the enabling or disabling of the history; directing the history to the terminal, line printer, or a text file; and displaying selected portions of the history by supplying a time or entry number range. When enabled, the session history should be readily available during a processing session.

An image history documents the processing performed on a specific image. Items that should be included in this history are (1) the names of the functions used to create or modify the image, (2) the parameter values used for each function, and (3) the date and time each function was used. The image history should always be enabled and available for user review. The user should not be responsible for updating or otherwise maintaining the history information. Each application function should update the history as needed. The history information must be closely coupled with each image. This could be accomplished by storing it as label records in the image file or in a separate file that is automatically associated with each image.

Programming Environment. If in-house algorithm development and enhancement are anticipated, an evaluation of the system's programming environment should be conducted. Of primary concern is the availability of callable subroutines to perform typical processing tasks (see Table 4).

One of the more common, but intricate, tasks is that of file handling. As an example, before an existing image can be read, several intermediate operations must be performed. These include (1) checking for the file's existence, (2) checking the location of the file (i.e., online or offline), (3) checking for proper data type, (4) checking validity of user requested image area and bands, and (5) opening the file for a specific type of access (random or sequential). If an output image file is to be created, a check should also be made for sufficient disk space.

A minimal number of subroutine calls should be required to perform a fundamental operation. For example, gaining access to an existing file should only require a call to one support routine which performs all the intermediate operations described above. This not only simplifies the programming task but also allows for low level software changes to be made without affecting the applications software.

TESTING PROCEDURE

Once the requirements and evaluation criteria are determined for the image processing system, the next step is to perform the actual testing and results verification. Although an exhaustive testing operation may not be feasible, detailed testing of specific aspects of the system may still be desirable. In the following discussion, it is assumed that a detailed testing of the entire system will be performed.

DATA SET COMPILATION

A basic set of testing data is essential for testing the system. Some of these data may be from external sources such as tape. Other data may be generated within the system itself. The main point is that the data be representative of the data that are projected for typical use on the system. A multispectral image of byte data is a fundamental data set that is quite useful. The number of bands and the line and pixel dimensions of this image should be sufficiently large to allow evaluation of documented system limits. Subsets of this image may be converted to various data types (integer*4, real*4, etc.) and used to test functions that process specific data types.

Operations such as grey level mapping and band ratioing may result in division by zero. If the software is not designed to detect this condition or other arithmetic errors, premature program termination will likely result. Therefore, explicit testing for such errors should be done to obtain an overall indication of how error proof the software is. An image consisting of uniform values is useful for trapping such arithmetic errors. Another data set that is useful for testing purposes is a "wedge" image which contains progressively increasing pixel values in the line and pixel directions. Such an image is helpful for testing operations such as histogramming or statistical calculations because the data characteristics are known. Imagery containing radiometric anomalies (e.g., striping) is necessary to test functions that perform radiometric correction.

Specific types of data are required to evaluate specific system capabilities. In order to evaluate the data ingest capabilities of the system, data on tape in various formats (e.g., Universal, Telespazio, Digital Elevation Model) is required. If a geographic information system is integrated with the image processing system, maps and associated map information such as surface data will be required. Lastly, data are needed to test interfaces between existing subsystems. This type of data includes control points, statistics, and polygon vertices and will generally be created on the system itself.

Table 4. Capabilities of Fundamental Programming Support Routines

- Parameter prompting and retrieval
- Image file handling (open, close, read, write)
- Ancillary file handling
- Output of standard error messages
- Session and image history handling
- Mathematical and statistical processing
- Input and output to display terminals

EVALUATION CHECKLIST

An evaluation checklist provides an orderly and consistent means for testing the system. The checklist should be general enough so that it can be used for evaluating a majority of the individual applications functions. For the LAS evaluation, the checklist was divided into three general categories: (1) documentation, (2) functionality, and (3) user friendliness. Within each category, the specific evaluation criteria previously mentioned were listed in concise statement form. For example, the statement, "Facilitates spectral and spatial subsetting," was used to verify that a function allowed the appropriate subsetting. In addition to the actual evaluation criteria, the checklist should provide space for any extensive explanations or comments.

INDIVIDUAL FUNCTION TESTING

After compiling test data and an evaluation checklist, testing of the individual applications functions can commence. The flow of testing is facilitated if the functions are prioritized into meaningful categories. A reasonable prioritization is data input and output operations, fundamental image processing operations (e.g., copying, addition, multiplication), and advanced image processing operations (e.g., pattern recognition, Fourier analysis, linear transformations). This breakdown is reasonable because data must be available on the system before further testing can occur and simple, more frequently used operations should be evaluated before the less frequently used but more complex ones.

An effective procedure for testing a given function is to (1) check online and offline documentation, (2) perform preliminary testing, and (3) perform detailed testing. The documentation should be checked for consistency and content based on previously mentioned criteria. Preliminary testing includes checking for blatant errors (e.g., function does not work at all), documented processing limits, parameter validation, spectral and spatial subsetting, input data set existence, and availability of sufficient system resources (e.g., disk space). The detailed testing includes algorithm verification as well as checking for agreement between documented and issued error messages, acceptable processing speed, consistent responses to prompts, updating of session and image histories, and remaining user friendliness features deemed to be important.

Because the most basic requirement of a system is its software functionality, particular attention needs to be focused on algorithm verification. This involves first running each function and obtaining the resulting outputs. Additionally, pixel value listings, statistics, histograms, and other useful information may be subsequently obtained. Verification of the output results may then be performed using the available information in conjuction with hand calculations or comparison to similar outputs from an existing image processing system.

MACROMODULE TESTING

Although all the individual components of a system may perform properly within their own limitations, they may not interface to other components when run in combination. Therefore, macromodules that contain several individual system components to be run serially should be constructed to test the interfacing of the individual functions and subsystems. File incompatibilities and size limitations are two potential types of interfacing problems that may arise. Size limitations include differences in spatial and spectral limitations between functions as well as differences in the number of values handled by each function. For example, a statistics file may contain 99 classes but the classification software may only handle 64 classes. File incompatibilities include data type mismatches for image files and differences in record sizes and record formats for non-image files.

As with testing individual applications functions, macromodule testing logically begins with the simplest ones and proceeds to the more complex. An example of a fundamental macromodule would be one to test the system's data set cataloguing (catalog management) capabilities. Testing of such a macromodule should precede testing of more complex ones for operations such as geometric registration, classification, and Fourier analysis.

The macromodules should represent the real end-to-end processing expected to be performed on the system. A good way to constuct the macromodules is to envision the processing needs for a given task or project. An example of a project would be to create a land-cover classification using digital imagery and then check the accuracy of the classification against an existing land-cover map. In the LAS evaluation, eleven such macromodules were constructed and tested (Table 5). Of the 224 functions in the LAS, the eleven macromodules used 98 in various combinations.

DISCUSSION

The evaluation criteria and testing procedures described herein were used to develop a formal audit plan for evaluation of the LAS. This evaluation took place over a two-year period by two full time personnel with experience in using other image processing systems. In addition to the image applications functions, a display system, a statistical analysis system, and a digitizing system were evaluated (Newcomer and Kovalick, 1985). Certain aspects of the evaluation including documentation, user friendliness, and image processing capabilities warrant further discussion.

Regarding documentation, the combined effort of audit and development personnel helped ensure its consistency and content. Because several people worked in developing the LAS, within-function inconsistencies, such as differences between online and offline documentation, existed. Review of all documentation by audit personnel helped remove these inconsistencies. The content of the documentation also benefited. That is, the programmers provided the needed algorithm descriptions and audit personnel helped relate the processing to the anticipated science applications.

The degree of user friendliness imposed on the LAS software was relatively high. This lengthened the development time but ultimately improved the usability of the system, especially by novice users. Certain facets of user friendliness such as parameter value validation, understandable error messages, and handling of spectral and spatial subsets received particular attention. Retrospective review of the evaluation revealed that other aspects such as paremeter naming conventions and verification of sufficient system resources should have received further scrutiny.

TABLE 5. MACROMODULES USED FOR THE LAS EVALUATION

- Data transfer
- Preprocessing
- Geographic image registration
- Data transformation
- Rasterization of digitized map data
- Supervised classification
- Unsupervised classification
- Spatial and frequency feature extraction
- Statistical analysis system interface
- Display subsystem
- Catalog manager and tape library

In terms of image processing capabilities, critical consideration was given to proper output results and handling of different data types. Due to explicit requirement specifications in the early stages of development, minimal problems were found in these areas. Subsequent use of the LAS indicated higher priority should have been given to processing efficiency. For example, the output of a processing status message for each image data line processed improved the user friendliness of the system, but image processing speed was significantly compromised. A new implementation of the processing message in which the message appears at user specifiable time intervals has been found to significantly improve processing performance.

Because intense usage of the display system was anticipated for the LAS, this area of the system services received a thorough evaluation. As stated, the requirements of the display system were met. However, it became clear after subsequent use of the display software that some functions should have been more modular. For instance, selection of control points on an image requires the display of different subsets, the use of different intensity mappings, and display of different areas at selected magnifications. It is very convenient if each of these operations is provided by a separate program which can be run in any combination rather than providing them in one large program. When a number of capabilities are included in one large program, this often necessitates moving through the program via specific processing paths or hierarchies; this results in decreased flexibility and inconvenience to the user.

The use of two individuals for the actual LAS evaluation proved quite effective. The different perspective provided by each person helped ensure a more complete evaluation. In particular, the construction of a comprehensive checklist benefited from the diversified experience of the personnel. Our experience has shown that derivation of an appropriate checklist is an iterative process. After an initial checklist is drafted, testing of a representative sample of functions should be performed to verify the completeness of the checklist.

The LAS evaluation was facilitated by access to existing image processing systems for results comparison. Considerable time was saved by not having to verify all output results by means of more lengthy processes such as hand calculations. That is, when the output results agreed, the LAS results were accepted as correct because it would be improbable that both sets of results were incorrect in the same exact manner. Notably, when output results did differ, further examination revealed that the LAS results were not always the ones that were incorrect.

CONCLUSION

Because of the expense of purchasing and maintaining an interactive multispectral image processing system, the acceptance of a proposed system should be contingent on a thorough evaluation of it. Although an evaluation could conceivably be determined by examining the appropriate documentation, this approach would not provide information on aspects such as ease of new function integration or processing performance. Some degree of actual testing is therefore essential. Our experience with the LAS evaluation has identified four system elements that warrant evaluation. These are documentation, user friendliness, processing capabilities, and system services.

The authors feel that the evaluation criteria and testing procedures for the LAS were quite effective. The use of testing personnel with experience in analysis of remotely sensed data provided a measure of quality control not likely to be achieved by the programmers themselves. In this regard, the construction and testing of macromodules was a significant advantage of our evaluation approach because individual applications programmers are probably not overly concerned with testing several functions in series. The availability of other image processing systems was another advantage of our evaluation approach. Knowledge of the strong and weak points of these systems greatly facilitated the development of an evaluation checklist.

It is acknowledged that the LAS evaluation was performed on a system that was under development and that such an extensive evaluation may not be warranted or desired on an existing, vendor supplied image processing system. Testing of specific aspects of a system or of a subset of the applications functions may still be desired. The authors believe that the information presented here can be applied to this type of evaluation as well as a more extensive one.

REFERENCES

Bracken, P. A., 1983. *Manual of Remote Sensing*, American Society of Photogrammetry, Falls Church, Virginia, pp 807–839.

ESL, 1983. *IDIMS Functional Guide*, Volume 1, Sunnyvale, California.

Goetz, A. F., G. Vane, J. E. Solomon, and B. N. Rock, 1985. Imaging spectrometer for Earth remote sensing. *Science*, Vol. 228, No. 4704, pp. 1147–1153.

Greenlee, D. D., and H. L. Wagner, 1982. Evaluation of a microprocessor-based Remote Image Processing System for analysis and display of cartographic data. *Proceedings of 5th International Symposium on Computer Assisted Cartography*, Crystal City, Virginia, pp 357–365.

Haralick, R. M., and G. Minden, 1978. KANDIDATS: an interactive image processing system. *Computer Graphics and Image Processing*, Vol. 8, No. 1, pp. 1–15.

McMurtry, G. J., F. Y. Borden, H. A. Weeden, and G. W. Petersen, 1977. *The ORSER system for processing and analyzing Landsat and other MSS data*. Tech. Rep. 7–77, ORSER, Penn. State University, Univ. Park, Pennsylvania.

Newcomer, J. A., and W. K. Kovalick, 1985. *LAS final audit report*. Performed under NASA contract NAS 5-28200, Science Applications Research, Lanham, Maryland.

Spencer, P. W., 1974. *LARSYS users manual*. LARS, Purdue University, West Lafayette, Indiana.

Vane, G., M. Chrisp, H. Enmark, S. Macenka, and J. Solomon, 1984. Airborne visible/infrared imaging spectrometer (AVIRIS): an advanced tool for Earth sensing. *Proc., International Geoscience and Remote Sensing Symp.*, Strasborg, France, pp. 1–7.

(Received 13 August 1986; revised and accepted 18 March 1987)

CALL FOR PAPERS

Sixth Thematic Conference on Remote Sensing for Exploration Geology

Houston, Texas
16–19 May 1988

The conference will emphasize applications, technology, and economics and will be designed to benefit both small independent operators and large corporate users of remote sensing. The technical program will be heavily oriented toward practical applications of remote sensing for mineral and hydrocarbon exploration, with special emphasis on the cost benefits that can be realized from use of remote sensing. Specific topic areas include

- Introduction to Remote Sensing: Technology and Data Availability
- Remote Sensing for Independents
- Frontier Exploration
- Exploration in Mature Basins
- Direct Detection of Hydrocarbons and Geobotanical Techniques for Exploration
- Offshore Applications
- Coal and Mineral Exploration
- Engineering Logistics and Environmental Applications
- Advanced Image Processing
- Remote Sensing for the 1990s

Those wishing to make a poster presentation should submit a 200- to 500-word summary no later than *12 October 1987* to

Dr. Robert H. Rogers
Environmental Research Institute of Michigan
P. O. Box 8618
Ann Arbor, MI 48107-8618
Tele. (313) 994-1200, ext. 3382