2

TEC-0004

US Army Corps
of Engineers
Topographic
Engineering Center

Hyperspectral Image Exploitation SBIR Phase I Final Report

T
E
C

SETS Technology, Inc.
300 Kahelu Avenue, Suite 10
Mililani, HI 96789

January 1992

Approved for public release; distribution is unlimited.

U.S. Army Corps of Engineers
Topographic Engineering Center
Fort Belvoir, Virginia 22060-5546

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
January 1992

3. REPORT TYPE AND DATES COVERED
Final Report September 1988 - March 1989

4. TITLE AND SUBTITLE
Hyperspectral Image Exploitation
SBIR Phase I Final Report

5. FUNDING NUMBERS
DACA72-88-C-0003

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
SETS Technology, Inc.
300 Kahelu Avenue, Suite 10
Mililani, HI 96789

8. PERFORMING ORGANIZATION REPORT NUMBER
TEC-0004

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army Topographic Engineering Center
Fort Belvoir, VA 22060-5546

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
TEC-0004

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The overall objectives of Phase I were to carry out the necessary research and design and to develop the techniques needed to demonstrate the feasibility of an overall concept and approach to the development of a hyperspectral image processing system. The development of a working system would then be the goal of Phase II. All of the main objectives of Phase I were accomplished to one degree or another. It was determined that a next generation Hyperspectral Image Processing system could be designed and built which would allow the customer to evaluate the current state-of-the-art in hyperspectral data processing and to get an idea on how it could be useful in their environment.

14. SUBJECT TERMS
Hyperspectral, Spectroscopy, Computer Image Processing, Remote Sensing

15. NUMBER OF PAGES
24

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UNLIMITED

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

SBIR Phase I Final Report

Hyperspectral Image Exploitation

I. Summary

All of the main objectives of Phase I were accomplished to one degree or another. It was determined that a next generation Hyperspectral Image Processing system could be designed and built which would allow the customer to evaluate the current state-of-the-art in hyperspectral data processing and to get an idea on how it could be useful in their environment.

II. Objectives of Phase I

The overall objectives of Phase I were to carry out the necessary research and design and to develop the techniques needed to demonstrate the feasibility of an overall concept and approach to the development of a hyperspectral image processing system. The development of a working system would then be the goal of Phase II.

Seven technical objectives were defined for the Phase I work:

<u>Objective 1</u> **Up-date our existing survey and understanding of the state-of-the art of hyperspectral image processing.** This involves the review of software systems currently being used for hyperspectral data analysis and the determination of the features to be incorporated in the next generation system.

<u>Objective 2</u> **Conceptually design a next-generation integrated hyperspectral data processing system.** This involves the creation of a full conceptual design of the next-generation system, based on SETS, Inc. experience with existing systems and the ideas of SETS, Inc. personnel for the next-generation system.

<u>Objective 3</u> **Study user interfaces.** The most rapidly evolving component of high-powered workstation computing environments is the user interface. The visual inspection of hyperspectral imagery, spectra and derived data products is crucial to scene analysis, and today's window-based graphical user interfaces provide ideal tools for this task.

<u>Objective 4</u> **Research and design new backout techniques for atmospheric and environmental effects.** The components of the Earth's atmosphere, by absorbing and scattering electromagnetic radiation, have a profound effect on the spectra emitted or reflected from materials on the Earth's surface. These atmospheric effects vary with location, altitude, time, and the presence or absence of materials within the atmosphere, such as smoke. Before identification of target materials on the Earth's surface is possible, these atmospheric effects must be "backed out." Currently, however, there are no known effective techniques implemented that permit atmospheric effects to be backed out on a pixel-by-pixel basis.

<u>Objective 5</u> **Study techniques for performing sub-pixel de-mixing.** Because of the finite spatial resolution of imaging spectrometers, most pixels in a scene will have spectra that are generated by a mixture of materials. To identify targets in such scenes, techniques must be developed for identifying sub-pixel targets. This involves the de-mixing or de-convolving of the mixed spectra.

<u>Objective 6</u> **Study spectrum encoding and searching algorithms.** Further research is needed to identify analytic methods that can be used to efficiently search spectral libraries against hyperspectral remote data for the identification of both targets and backgrounds. Encoding algorithms also need to be studied to reduce the effective size of the hyperspectral data file to be processed for faster operation and better utilization of the computer hardware.

<u>Objective 7</u> **Explore the application of knowledge-based and artificial intelligence approaches.** The use of a knowledge-based system will better allow the use of remote sensing technology in the field by non-technical users.

III. Research Conducted

In Phase I, three specific areas of effort were defined under which these objectives would be pursued. These three areas were:

- Technology Development
- Technique Development
- Customer Evaluation

In this section, the research conducted for Phase I is described in separate sub-sections for each of these areas of effort, with additional reference to the technical objectives. In Section **IV. Results**, the findings of the research in these areas are combined into a comprehensive description of the hyperspectral cube processing system that will be implemented in Phase II.

III.1 Technology Development

Efforts in the area of technology development in Phase I focused on the conceptual design of a data processing system geared exclusively to the processing and analysis of hyperspectral image cubes. The hyperspectral cube processing system envisioned will include most standard image processing capabilities, most standard spectral processing capabilities, and a set of capabilities designed exclusively for the analysis of hyperspectral cubes. In addition, the system will contain an interface to a set of spectral libraries.

Phase I research and development efforts in the area of technology are detailed below.

III.1.1 Technical Survey

Hyperspectral cubes are a new type of data. The instruments that generate such data sets—imaging spectrometers—are in their infancy, so much so that it is doubtful whether any useful (i.e. calibrated, low signal-to-noise ratio) hyperspectral cubes are yet in existence. Software systems for analyzing this type of data are likewise only a few years old.

A hyperspectral data processing system can be defined as an integrated package of functions that is able to operate on image data as single-plane two-dimensional data, as single-spectrum (one-dimensional) data, and especially as three-dimensional data, where the third dimension contains spectral information keyed to the pixel locations in the component two-dimensional planes.

Many systems have been developed for the processing and analysis of two-dimensional image data. Likewise, many systems have been developed for the analysis of one-dimensional spectral data. But only a small number of true hyperspectral cube processing systems have been developed (though many image processing systems do have some subset of tools that can be used on multi-spectral or hyperspectral data). Most early hyperspectral cube processing systems were derived from image processing systems, though today an increasing percentage of such systems are being designed and implemented specifically for hyperspectral data.

SETS, Inc. attempts to continually stay abreast of developments in the field of hyperspectral cube processing systems. In the past, SETS, Inc. has subjected several of the best of these systems to detailed evaluations. For Phase I, SETS, Inc. personnel updated their existing survey of existing hyperspectral cube processing systems. The systems surveyed are discussed below.

- **GARP (Geosciences ARray Processing System)**, developed at the Planetary Geosciences Division at the University of Hawaii. Most of the early systems for processing hyperspectral data sets were derived from existing image processing systems and thus suffered from the relatively constrained thinking of the image processing world. A notable exception to this is the Geosciences ARray Processing (GARP) system, which was conceived and implemented in the early 80's for the exclusive purpose of analyzing and manipulating multi-dimensional cubes of data, specifically the hyperspectral cubes generated by the Near Infrared Mapping Spectrometer (NIMS) instrument of the Galileo mission to Jupiter (current launch date: late 1989). Several SETS, Inc. scientists were involved in the design and implementation of GARP, and this experience will add to the quality and utility of the system to be developed for Phase II.

GARP is an interactive software system for processing n-dimensional arrays of scientific data. A typical array consists of a set of images of the same spatial scene, with members of the set varying in observational parameters such as wavelength, phase angle, and time of observation. (Because of GARP's generalized design, other arrangements of dimensions can also be processed.) GARP provides a comprehensive set of functions for displaying, translating, analyzing, reducing, and geometrically manipulating such arrays. GARP uses a tile-based data format and operates under the UNIX environment, and supports color display on both the International Imaging System IVAS display processor and on color Sun monitors.

- **ISIS (Imaging Spectrometer Interactive System)**, being developed by SETS, Inc., USGS (Flagstaff), UCLA, and JPL, for the analysis of data from the Galileo NIMS instrument and other imaging spectrometers. This system is essentially a complete rebuild of GARP. For political reasons, the group of institutions developing ISIS chose not to utilize and build on GARP, and instead are developing a new system from scratch. The history of the development of this system clearly demonstrates why software should never be developed by a geographically, politically, and philosophically dispersed group of institutions. Despite its arduous development, ISIS already contains some good functionality, and it will acquire more in the coming years. ISIS will continue to be developed into the early 90's until, and perhaps beyond, Jupiter encounter in 1996. ISIS is being developed under the VMS environment, though the system is being implemented with future portability to UNIX in mind.

- **QLOOK (Quick Look)**, also being developed by SETS, Inc., USGS

(Flagstaff), UCLA, and JPL. QLOOK is an experimental system for the highly-interactive viewing and analysis of hyperspectral cubes. It is being developed in parallel with ISIS and is probably the prototype of what ISIS will evolve to in the future. QLOOK is being developed under a VMS environment, and early versions have been ported to UNIX machines.

- **WISP (Washington Image and Spectral Package)**, developed by Dr. John Adams, University of Washington, Seattle. WISP is a highly interactive system for the processing and manipulation of multi-spectral and hyperspectral data cubes. WISP is implemented on Symbolics 3600 computers under the Genera operating system. WISP implements advanced tools for the analysis of spectral mixtures (the Mixing Model) and for the viewing of multi-dimensional data (the Data Viewer).

- **IRAF (Image Reduction and Analysis Facility)**, originally developed by the Kitt Peak National Observatory and now being developed by the National Optical Astronomy Observatories (NOAO) in Tucson, Arizona. The IRAF system is a general-purpose image processing system with graphics applications. Although many of the functions which comprise this system are directed toward two-dimensional astronomical image processing, the basic system is designed around the "image cube" concept. Thus, those functions that are not currently set up to access generalized three-dimensional data sets could in most cases be modified to do so. IRAF operates under both the UNIX and VMS environments.

- **AIPS (Astronomical Image Processing System)**, developed by the National Radio Astronomy Observatory (NRAO) in Greenbank, West Virginia. AIPS is an interactive software system designed primarily for processing three-dimensional cubes of data obtained on astronomical objects. AIPS contains many of the generalized software tools needed in an image cube processing system. The data format used by AIPS is sufficiently flexible that it could also be used to operate on other data sets. AIPS operates under the UNIX and VMS environments.

- **SPAM (SPectral Analysis Manager)**, developed by the Jet Propulsion Laboratory (JPL) in Pasadena, California. SPAM is an interactive software system designed for processing data obtained with NASA's Airborne Imaging Spectrometer (AIS) and Advanced Visible and InfraRed Imaging Spectrometer (AVIRIS). SPAM operates under the UNIX and VMS environments.

- **TDRS (Taurus Data Reduction System)**, developed by Dr. Joss Bland, University of Sussex, England. TDRS is not so much an image cube processing system as it is a collection of image cube processing functions—there is no system executive that controls access to the functions. TDRS was designed primarily for processing observations obtained with the TAURUS interferometer. However, the data format and the available functions are sufficiently generalized that, following the development of a suitable executive, TDRS could also be used to operate on other types of data. TDRS operates under the VMS environment.

III.1.2 Testing of Existing Systems

During Phase I, SETS, Inc. has continued its in house study of several of the key hyperspectral data processing systems. In particular, we have worked extensively with **GARP**, **ISIS** (the system being developed for NASA), **QLOOK** (also being developed for NASA), and **WISP** (being developed by SETS, Inc. and the University of Washington). When studying these systems, SETS particularly looked at:

1. The systems' general utilities for working with hyperspectral data.
2. Algorithms which are useful for processing in image space, the spatial vs spatial domain.
3. How these systems incorporated the use and handling of spectral data.
4. Ways to efficiently store and retrieve data.
5. The advantages and disadvantages of their various user interfaces.
6. What features were incorporated or not incorporated to make the system more portable.

GARP is the most general of systems which were tested: it allows operation on data sets containing up to 10 dimensions and incorporates most of the features required in a basic hyperspectral data processing system. However, GARP has several main limitations: 1) an inefficient method of data storage when operating in the spectral domain, 2) an awkward user interface, 3) the inability to work with single dimensional data (i.e. a single spectra), and 4) a low degree of portability.

ISIS is currently being developed by SETS, Inc., USGS (Flagstaff), and UCLA for the analysis of data from the Galileo NIMS instrument and other imaging spectrometers. The development of this system clearly demonstrates the problems inherent in developing software by teams which is dispersed both geographically and politically. This system, however, has good potential, but is not due for completion untill 1996. **ISIS**, by design, has several key features: 1) a data format which will allow quick and easy access to the data, 2) spectral library capabilities, and 3) the ability to operate in several modes of operation including batch, command, and menu modes. The limitations as we see it are: 1) its estimated completion time, 2) the use of the TAE user interface, and 3) the development of ISIS specifically around a microVax workstation and I^2S

monitor.

QLOOK, which is being developed by the same team as ISIS, is only one part of a total hyperspectral data processing system. QLOOK, at this point, is an experimental system for highly-interactive viewing and analysis of hyperspectral cubes. QLOOK will gradually become an integral part of ISIS and shows the utility of having a quick and easy of viewing the entire data set. Another example of this type of interface is the data cube display system developed for the PIXAR display at ETL.

WISP, which is currently being developed by SETS, Inc. and the University of Washington, is a highly interactive system for the processing and manipulating of multi-spectral and hyperspectral data sets. It has two main features which should be incorporated into the hyperspectral processing system being proposed: 1) the Mixing Model approach to the mixed pixel problem and 2) the Data Viewer for the viewing of multi-dimensional data space.

III.1.3 Computer System Evaluation

SETS, Inc. has completed its preliminary evaluation of computer systems, even though it is proposed that this work be continued during any follow on phases, so that the system being developed will utilize new and better platforms as they are developed. The criteria used in evalution of the existing systems were:

1. The operating system should be widely used on a number of different computer systems and architectures.
2. The system should allow programing in the C programming language.
3. The system should be able to contain both large amounts of memory and data storage.
4. The system should support a bit mapped monitor, with at least 256 colors, which allows the use of a windowing environment.
5. The system should have high rate for the number of floating points instructions per second.
6. The system should be able to support external monitors if desired, such as a I^2S or a PIXAR.
7. The system should be capable of acting as a front end to a parallel processor.

8. The computer manufacturer should be a stable, well established firm.

Several computer systems were evaluated in Phase I:

- A **Sun 4/280** color UNIX workstation with an integral "applications accelerator/display processor" known as a TAAC system. This is currently the main development platform at SETS and is the system on which the most advanced version of GARP resides.

- A **Sun 3/60** color UNIX workstation. This system has been used to implement a version of GARP that utilizes color display in Sunview windows (instead of relying on an external International Imaging Systems IVAS display processor, as previous versions of GARP have). This system was also used to study the Sunview version of SPAM and is also utilized as an ARC/Info GIS workstation.

- A **Sun 3/180** UNIX workstation with an IVAS display processor. This system, located at the University of Hawaii, is the first workstation to which GARP was ported.

- A **VAXstation II/GPX** color VMS workstation with an IVAS display processor. This system is the chosen platform for the development of the ISIS software system and for QLOOK.

- A **Symbolics 3650** artificial intelligence workstation with a 24-bit color system. This is the system on which the WISP is implemented.

- A **Hewlett-Packard Vectra RS/20** 80386-based machine running MS-DOS and Windows is currently under evaluation for another project. This machine may also be utilized to demonstrate the portability of Phase II developed software to powerful 80386-based UNIX workstations.

- **Connection Machine** Preliminary testing was conducted with the Connection Machine to ascertain its utility as a computer engine for hyperspectral data processing systems. Testing was conducted by implementing WISP on a Symbolics attached to a Connection Machine I and by implementing some stripped down GARP routines on a VAX attached to a Connection Machine II. Testing is not complete, but preliminary analysis suggests that the Connection Machine has great promise as a part of hyperspectral data processing systems, particularly if I/O bottlenecks can be eliminated.

Based upon these criteria, SETS, Inc. chose to develop the proposed hyperspectral data processing system on a color Sun 4 microcomputer utilizing a TAAC image display and application accelerator system.

III.1.4 Study of User Interfaces

Significant research on user interfaces was conducted focussing on windowing systems and standards (e.g. X-Windows, NeWS), Graphical User Interfaces (e.g. Open Look, DECwindows), and modes of user interaction (e.g. command mode, menu mode, batch mode, highly-interactive mode).

Specific activities included the study of windowing systems and GUI's through reading professional journals, written and verbal contact with other computer scientists, and interviews with the sales and technical personnel of computer and software firms; direct hands-on experience with several Graphical User Interfaces, including DEC's UIS, Sun's SunView, Microsoft's Windows, Apple's Macintosh environment, and Symbolics' Genera environment; direct hands-on experience with several software systems that implement the various modes of user interaction, including GARP, WISP, ISIS, QLOOK, and CHAPS; and the recollection and reevaluation of previous experiences in the design, development, maintenance, and use of a variety of user interfaces.

The user interface defines how the computer and the user interact. Essentially, the user interface defines the "personality" of the computer in its interactions with humans. Prior to the advent and popular use of bit-mapped screens, mice, icon-based screens, and pull-down menus, computer user interfaces were implemented on standard 80 column by 24 line ASCII terminals. Users commonly entered their desires by typing terse and often arcane commands at a command line prompt or, in more "friendly" systems, by entering selections from menus and filling in blanks on function prompt screens.

In today's world, the implementation of a user interface on a standard 80 column / 24 line screen is as outdated as writing computer programs on punched cards. Thus our research focused entirely on the study of Graphical User Interfaces, or GUI's, which are user interfaces implemented in computing environments characterized by bit-mapped screens, mice, icons, and pull-down menus.

GUI's as we know them today were originally developed by XEROX researchers and were first implemented on XEROX artificial intelligence workstations. The first popular GUI was implemented on the Apple Macintosh. Since that time, Sun, DEC, IBM, and other manufacturers have made major commitments to developing and maintaining GUI's on their systems.

Every GUI has a distinctive "look and feel," which defines how it interacts with the user. Elements of "look and feel" include how a window is activated (e.g. by moving the cursor into it or by clicking in the window), how menu choices are made (e.g. by simply moving the cursor or by clicking on the desired choice), how many buttons are on the mouse and what they mean in different contexts, and so on.

Recently, several major players in the computer industry have made moves to develop "standardized" GUI's that can be run not only on their machines but on other machines as well. DEC is promoting DECwindows, Sun is promoting Open Look, IBM is promoting Presentation Manager, HP is promoting NewWave, and so on. DECwindows and Open Look have been designed to run on top of a standardized library of windowing primitives known as X-Windows Version 11, Release 2. Basing these GUI's on X-Windows makes them portable to any other machine on which the X-Windows library has been implemented.

Open Look and DECwindows are also designed to operate by '"network extensible," meaning that a user on a local bit-mapped workstation can log onto a remote bit-mapped workstation and utilize the GUI on the remote workstation just as if it were running on the local machine. Without network extensibility, this arrangement is impossible because it requires the continual transmittal of large bit-mapped screen images between the computers. With network extensibility, the contents of a bit-mapped screen display are "tokenized" or symbolized so they can be passed efficiently from one machine to the other. The local machine interprets tokens sent by the remote machine and implements them on the local bit-mapped screen.

Several modes of user interaction are possible in hyperspectral processing systems, allowing the user to control the flow of data processing in several ways. In **menu mode**, the user is presented with a hierarchy of menus. Selection of a menu item takes the user either into another menu or into a **function prompt screen**, where parameters are entered for particular functions. In **command mode**, users can bypass the menu system and enter particular function prompt screens directly with a short (1-3) letter command. In **batch mode**, the user fills out the parameters for a sequence of individual functions and then submits the list of commands and parameters as a batch job, to be run in background. In **highly-interactive mode**, the user interacts in a highly-interactive mode with a visual display, utilizing the mouse and simple commands to control the display and the processing of data.

As indicated in the Results section below, it was decided to implement the Phase II system utilizing either DECwindows or Open Look, depending on which GUI looked most promising at the time programming begins. Both of these systems are based on the X-Windows library of windowing primitives, and both offer network extensibility (see Results section for more explanation of this term).

III.1.5 Exploration of the Use of Artificial Intelligence

Some of the analysis of computer hardware and hyperspectral cube processing systems was carried out on a Symbolics 3650 artificial intelligence workstation. None of the code analyzed on this or other machines, however, can really be called "artificially intelligent." SETS, Inc. scientists consider this term to imply the use of expert systems or other systems that utilize decision-making strategies based on the past experiences of the software system, or of the remote sensing scientists who provided the initial database of knowledge upon which the system is based.

As far as SETS, Inc. scientists know, there are no existing expert systems or knowledge-based systems in the field of hyperspectral remote sensing. Such systems do exist in other fields, such as chemical engineering and medicine, and it is these existing systems that should be studied and modified for use in remote sensing. Due to lack of time and resources, these studies were not conducted in Phase I.

III.1.6 Conceptual Design

The conceptual design of the hyperspectral data system is described in section IV. Results.

III.2 Technique Development

Because hyperspectral data are a new and rich source of information remote sensing scientists have not yet learned efficient means for tapping the tremendous amounts of information latent in these data sets. New techniques and tools for deriving this information are being developed rapidly as the availability and understanding of these data sets grows.

Outlined below are the techniques and algorithms for processing hyperspectral data that were considered for Phase I. They are grouped into three major categories: atmospheric backout, sub-pixel mixture analysis, and spectrum feature characterization. Results under each of these categories are discussed here.

III.2.1 Atmospheric Backout

It is an ongoing task at SETS, Inc. to maintain an up-to-date review of the methods employed or proposed for performing an atmospheric backout of hyperspectral data. Methods reviewed to date are: 1) The flat-field correction method (developed at JPL); 2) the logarithmic-residuals method (developed at CSIRO, Sydney); 3) the in-scene calibration standard (used by a variety of researchers); and 4) the band-depth / equivalent density method (developed by SETS, Inc.).

The band depth / equivalent density method was developed during the time-frame of the Phase I effort and was funded in part by that contract. Attached is a detailed description of the band depth / equivalent density method. This report also includes a review of other atmospheric correction techniques.

III.2.2 Sub-pixel Mixture Analysis

As part of the Phase I effort, a review was conducted of the methods presently available for identifying the contributions to a given pixel element. These techniques may be divided into two categories: 1) Methods that define the features in the spectrum (such as absorption band presence, position, and depth); and 2) methods that break a pixel spectrum into spectra corresponding to each of the contributing materials in the pixel.

Under the first approach, the problem remains of relating the spectrum features to materials on the ground (assuming that other contributing elements such as detector system and atmosphere have previously been accounted for.) The second approach, while providing a full deconvolution of the data to the actual contributing spectra, must assume some model of how the data are convolved; this can require an understanding of surface (e.g. topography) as well as spectral effects.

It has been shown previously (e.g. Adams et al., 1986) that it is possible, with well-constrained datasets, to identify the sub-pixel components contributing to the signal received from a pixel. However, this is not a simple problem to solve. Before the question of target detectability and identification can even be approached, the effects of system response, system noise, and the intervening atmosphere must be well understood and accounted for.

Both of the above approaches (feature identification and spectrum modeling) have been explored under the present contract. Results of the spectrum modeling work are described here. Results of the research into spectrum feature identification methods are described below under the Spectrum Feature Characterization sub-task.

Spectrum Modeling: There has been only one method identified which adopts the second approach to pixel de-mixing. This is the linear mixing model technique originated by Adams et al. (1986). However, a similar, though less comprehensive, method has been developed at JPL. This is the method of binary encoding and spectrum matching (using the Hamming distance measure for binary data). This encoding and alarming method has been implemented as part of the in-house hyperspectral data processing system.

The linear mixing model method is currently resident on a Symbolics computer in the LISP programming language; however, the technique may also be implemented under other configurations. As part of the present contract, this model has been implemented in-house under the Symbolics environment; a coordinated effort of development and testing of the model has been initiated.

The mixing model may be employed in two configurations. Under the first, spectra are systematically (although manually at present) extracted from the scene until all spectral contributions have been identified. In this mode, although the outcome is comprised of complete spectra, there is still the problem of relating the spectra to materials on the ground. Again, if the data have not been calibrated to physical units, then drawing the spectra/target relationships may be difficult to impossible to do with any certainty.

The second mode of use of the mixing model selectively extracts spectra from a database of spectra to construct the spectral contributions to the remote dataset. This method has not to date been employed, strictly because of the difficulty at present of relating laboratory spectra in a database to the spectra in a remotely obtained dataset. Work has only just begun on determining the criteria required to complete the link between the spectra in a database and the spectra in a remotely obtained hyperspectral dataset.

III.2.3 Spectrum Feature Characterization

Three of the methods that have been developed for identifying spectrum features have been implemented as part of the in-house hyperspectral data processing system. These are the method of convex hulls originated by Green and Craig (CSIRO, 1985), the Gaussian band-fitting method, and the band-depth mapping method originated by Singer and Blake (unpublished).

An important result of studying all three of these methods is the realization that the data being analyzed must be in physically meaningful units (e.g. reflectance). If the data have not been calibrated to some recognizable system of measure, then any "bands" or "features" that are identified may not reliably be related to the target, but may be merely artifacts of the system itself, or of the atmosphere. A "feature" in the signal cannot be considered a spectral "band" unless the data are in spectral units.

III.3 Customer Evaluation

Customer evaluation for Phase I includes the following components:

- Discussions with the customer of the overall desired project goals.
- Installation of a first generation image processing system at the

customer's base of operation.

- Training of the customer in the use of the generic system.
- Participation in several customer (at least simulated) applications using existing data sets to gain hands-on operational experience.
- Recording the customer feedback for use in the system design.

III.3.1 Review Project Goals and Objectives

During Phase I, SETS, Inc. maintained a close relationship with the customer to ensure that the research and design work being performed would result in a system which would better meet the customers needs. From this review, SETS, Inc. determined that the customer would use the system to evaluate if hyperspectral data processing would be useful in battle field target identification. This would be accomplished by identifying targets from known spectral signitures. Therefore, the system to be developed should incorporate the utilization of a spectral library for identification and discimination of target and background materials.

III.3.2 Installation of a Prototype Hyperspectral Image Processing System

The installation of the 1st generation hyperspectral data processing system was done in early April. The actual installation went fairly well, and no major setbacks were encountered. This system will be used by the customer to better understand the current state-of-the-art in hyperspectral data processing and to better interact with SETS, Inc. during the Phase II development.

III.3.3 Customer Training on the Data Processing System

On site customer training was done after the installation of the hyperspectral data processing system. This training went well and brought out further areas of work for both the customer and SETS, Inc.

IV. Results

This section summarizes the conclusions from the research conducted in Phase I. The results are presented in the form of a complete conceptual design of the system to be implemented in Phase II, including descriptions of the software system itself and its capabilities.

IV.1 The Design of the Software System

IV.1.1 Hardware

The system will be developed on a color Sun 4 Microcomputer utilizing a

TAAC image display and application accelerator system. The system will utilize the TAAC system when it is available and will also be able to provide all critical capability when a TAAC system is not present.

The system will also be designed to interface with a Connection Machine for the performance of computationally-intensive applications, if the local host is connected directly to a Connection Machine. If a Connection Machine is not present, the system will utilize the host processor's capabilities for all computations.

The system will be implemented in a fashion such that it can easily be ported to other hardware platforms, if desired. A standardized operating system, standardized windowing primitives and toolkits, and standardized graphics packages will all be utilized to assure this ease of portability. The features of the system that may not necessarily be easy to port are the interface to the TAAC applications accelerator and the interface to the Connection Machine. Portability of the TAAC code will be assured if porting to another model of Sun workstation that supports the TAAC. Portability of the Connection Machine interface will be assured if porting to an ULTRIX VAX, another Sun model, or to another machine supported by Thinking Machines, Inc. as a Connection Machine front-end.

IV.1.2 Operating System

The system will be developed to operate under a POSIX (standardized UNIX) operating system to provide maximal portability to other computing platforms. Initial development will take place under Sun's version of UNIX, SunOS, version 4.0 or higher.

IV.1.3 User Interface

It was decided to implement the Phase II system utilizing either DECwindows or Open Look, depending on which Graphical User Interface (GUI) looks most promising at the time programming begins. Both of these systems are based on the X-Windows system of windowing primitives, which is a de facto industry standard, and both offer network extensibility (see Research Conducted section for more explanation of this term). Among other advantages, the use of such a GUI will allow users to operate the system from a remote bit-mapped workstation, seeing exactly what a local user would see, but without actually running the software on the remote workstation. In a typical application, users will have a variety of windows open on their screen, containing color images, graphs of spectra, histograms, and other data, and command and input information and prompts.

The system will also be built with several modes of user interaction, allowing the user to control the flow of data processing in several ways. In **menu mode**, the user will be presented with a hierarchy of menus. Selection of a menu item will take the user either into another menu or into a **function prompt screen**, where parameters are entered for particular functions. In **command mode**, users can bypass the menu system and enter particular function prompt screens directly with a short (1-3) letter command. In **batch mode**, the user can fill out the parameters for a sequence of individual functions, and then submit the list of commands and parameters as a batch job, to be run in background. In **highly-interactive mode**, the user interacts in a highly-interactive mode with a visual display, utilizing the mouse and simple commands to control the display and the processing of data. Finally, the system will implement a "**co-function**" capability, allowing users to break in the middle of any function prompt screen, perform another function, and then return to exactly where they left off. This last capability has great utility and is a favorite with experienced users of systems in which it is implemented.

IV.1.4 Graphics Software

The system will utilize the Graphical Kernel System (GKS) for graphics primitives. GKS is an industry standard for graphics primitives, allowing the implementation of 2-D screen graphics in a fashion that is easily portable to other architectures.

IV.1.5 Programming Tools

The system will be built and maintained utilizing state-of-the-art programming tools, including a CASE (Computer-Aided Software Engineering) system, an automated software build system, a source code control system, and advanced debugging tools. CASE provides an overall environment for advanced software engineering. Automated software build systems allow software to be rebuilt again and again in the most efficient way possible, avoiding the unnecessary compilation and linking of unmodified code, and keeping track of complex code dependencies. A source code control system allows numerous programmers to work on the same software system at once without the duplication and/or canceling out of efforts often found in multiple-programmer situations. Debuggers allow programmers to take code apart piece by piece and correct it if it is malfunctioning.

IV.1.6 On-line Help

The system will contain a two-tier help system, the first level giving a short statement of help information, the second giving pages of help, if necessary.

IV.1.7 User Documentation

The system will be delivered with a comprehensive **User's Manual**, explaining in detail how to use the system, and describing each function, the function's arguments, and how the function works.

IV.1.8 Data Handling

The system will be capable of handling datasets of any size, limited only by available disk space. The data can be configured to be one, two, three, four, five, or six dimensional. The data can be in any of eight data types—byte, unsigned byte, short integer, unsigned short integer, long integer, unsigned long integer, floating point, and double precision. Data from a variety of sources will be handled via translation routines that will translate the foreign data formats into a native data format.

IV.1.9 Single Spectra Analysis Sub-system

The system will contain a graphical environment for the analysis of a single spectrum or several spectra at a time. The source of these spectra might be a hyperspectral data cube the user is working with in another part of the system, or they might have been pulled in from a spectral library. A variety of spectral analysis functions will be available in this environment, including functions for comparing a given spectrum against the spectra in a spectral database.

IV.1.10 Highly-Interactive Cube Viewing Sub-system

The system will contain a sub-system for the rapid viewing and preliminary analysis of hyperspectral cubes. This cube-viewing system will be modeled after several existing and highly useful systems.

IV.1.11 Spectral Library Interface

Several of the functions in the system will be capable of reading in spectra from a spectral library and using them for comparison or calculation with either the spectra in a cube or with single spectra in the single spectra analysis sub-system.

IV.1.12 TAAC Interface

The TAAC Application Accelerator consists of a two board set that resides within the cardcage bus of the Sun 4. The TAAC is "an ultra-high performance processor" with integrated full color display that provides the capability for greatly speeding up applications (between 20 and 100 times faster, depending on the application). The TAAC has two main capabilities: 1) As a display processor, it allows programs to display and manipulate 24-bit color images in a highly-interactive fashion. Images are displayed either within a window on the Sun screen or on a separate color monitor. 2) As an applications accelerator, the TAAC acts as a combined array processor, floating point accelerator, and graphics and image processor.

The software system to be developed in Phase II will be implemented to take advantage of the TAAC when it is present. In computationally intensive functions, the user will notice a significant decrease in execution speeds when the TAAC is present, but there will be no difference in functionality. In highly-interactive display functions, and particularly in the highly-interactive cube display subsystem, the user will likely have more functionality when the TAAC is present, simply because the TAAC allows highly interactive graphics and image display functions to be easily programmed.

IV.1.13 Connection Machine Interface

The Connection Machine is a Massively Parallel Processor (MPP) containing thousands of individual processors (typically 16K or 64K) that are designed to process large amounts of individual data elements in parallel. This "Single-Instruction Multiple-Data" (SIMD) architecture is ideally suited for the processing and analysis of hyperspectral data, where in a typical application, the same processing is performed on all the spectra in a hyperspectral cube.

The Connection Machine is designed to work with a front-end computer (a Symbolics, VAC, or Sun) that downloads code and data to the Connection Machine for processing. In effect, the Connection Machine can be thought of as a huge and powerful array processor to which computationally intensive tasks are downloaded by the host processor.

The software system to be developed in Phase II will be implemented to take advantage of a Connection Machine when one is available. The user will have the option as to whether to make use of the Connection Machine if it is available. In computationally-intensive functions, the user will notice a very significant decrease in execution times when the Connection Machine is present.

IV.2 The Capabilities of the Software System

Two categories of software capability are described here. The first category includes general cube processing functions, and the second involves specific cube processing capabilities, such as the capability for performing sub-pixel mixing analysis. Specific cube processing capabilities may utilize some of the general cube processing functions (e.g. some calibration procedures may be performed using simple mathematical functions), may utilize routines specifically designed for a given task, or may use a combination of available functions.

IV.2.1 General Cube Processing Capabilities

It was determined in the Phase I effort that the following **generalized** data manipulation and analysis functions would be part of a complete hyperspectral data analysis system:

File Manipulation Functions:

- List available image cubes and descriptive information from their labels
- Delete image cubes
- Create artificial image cubes
- Copy and rename image cubes
- List other ancillary files (e.g. picture files, spectrum databases)

Data Translation Functions:

- Translate image cubes to and from "foreign" formats
- Convolve image cubes to applications library format

Data Enhancement Functions:

- Filter image cube data (including Fourier transform functions)
- Smooth image cube data
- Perform edge enhancements of image cube data
- Perform image restorations of image cube data
- Find and remove surface of best-fit

Data Manipulation Functions:

- Mathematically manipulate image cubes and sub-cubes
- Expand and shrink image cubes
- Transpose image cubes to change axes orientation
- Coregister two image cubes to same coordinate system
- Visually examine and selectively alter image cube data
- Compute derivatives

Data Display Functions:

- Display, zoom, and pan image planes
- Display image cubes in highly interactive environment (e.g. TAAC)
- Create hardcopy versions of images, spectra, graphs, and numerical data
- Extract and display spatial profiles or "traverses" from image cubes
- Extract and display spectra from image cubes

Data Summary Functions:

- Compute and display histograms
- Compute and display statistical information on cube data
- Compute contour map of 2-D plane

IV.2.2 Specific Capabilities

It was determined in the Phase I effort that the following specialized data manipulation and analysis functions would be part of a complete hyperspectral data analysis system:

Data Calibration Functions:

- Remove solar component
- Compute and remove black body component
- Perform atmospheric backout
- Correct for system / instrument effects
- Compute and correct for photometric effects
- Normalize against input spectrum or data characteristics
- Perform geometric rectifications
- Compute and remove spectral continuum

Spectrum recognition and identification functions:

- Encode and decode spectral information in image cube data
- Search image cube against input spectrum (spectrum "alarming")
- Manipulate and analyze single spectra from cubes or spectrum libraries
- Compute spectrum diagnostics (e.g. absorption band position, depth, width)
- Search a cube for spectra meeting certain criteria defined by the user
- Sub-pixel demixing

Data Classification Procedures:

- Supervised and unsupervised unit mapping
- Polynomial fitting of cube spectra
- Mixing model-based

V. Estimates of Technical Feasibility

No technical problems have been identified that would prevent the implementation of the defined system under Phase II. However, it should be noted that the degree to which certain processing algorithms work, such as amtospheric correction, will be strongly affected by the data sets used (i.e. wavelength range, signal-to-noise ratio, and calibration).